VADNAI SZ[illegible]OLCS

COMMODORE 16

PROGRAMOZÓI ZSEBKÖNYV

VADNAI SZABOLCS

COMMODORE 16

PROGRAMOZÓI ZSEBKÖNYV

Felelős szerkesztő: Tarr Kálmánné

Lektor: Rácz Mihály

Kiadványmenedzser: Békés Tamás

Műszaki szerkesztő: Dévényi Erika

(C) NOVOTRADE RT.

A kiadásért felel a NOVOTRADE RT igazgatója

A kiadványt
az 1969 évi III.törvény 13. és a 18. első bekezdése alapján
tilos
bármilyen módon sokszorosítani és a sokszorosítást
terjeszteni

A szedési munkák
COMMODORE 64 számítógépen
EASY SCRIPT szövegszerkesztővel
MPS 801 mátrixprinterrel
készültek

ISBN 963 02 4603 1

AGROKÉPSZER GT.
OKISZ LABOR NYOMDA

TARTALOMJEGYZÉK

oldal

BEVEZETÉS 3

JELÖLÉSEK ÉS KONVENCIÓK 3

BASIC

1. SZINTAKTIKUS ELEMEK ÉS REPREZENTÁCIÓJUK 4
   1.1. Szintaktikus elemek
   1.2. Az ismertetett elemek memóriabeli reprezentációja
   1.3. A BASIC-Programok tárolásával kapcsolatos főbb memóriacímek 5
2. A BASIC 3.5 ALAPUTASÍTÁSAI 6
   2.1. Deklaratív utasítások
   2.2. Értékadó utasítások
   2.3. Programszervező utasítások
3. FÜGGVÉNYEK 7
   3.1. Matematikai függvények
   3.2. Karakter és string-kezelő függvények
   3.3. Memóriakezelő függvények
   3.4. Grafikus függvények
   3.5. Egyéb függvények
4. RENDSZERVÁLTOZÓK 8
5. GRAFIKUS ALKALMAZÁSOKAT SEGÍTŐ UTASÍTÁSOK
6. RENDSZER-FUNKCIÓK
7. PROGRAMOZÁSI SEGÉDESZKÖZÖK
8. PERIFÉRIÁK KEZELÉSE, ADAT BE ÉS KIVITEL 9
   8.1. Speciális I/O utasítások Programok be és kivitelére
   8.2. Az utasítások általános formája
   8.3. Az I/O utasítások által használt legfontosabb címek 10
9. BILLENTYŰZET 11
   9.1. Beolvasás billentyűzetről
   9.2. A billentyűzet működésével kapcsolatos főbb memóriahelyek
   9.3. A billentyű-mátrix
   9.4. A funkcióbillentyűkhöz alapértelmezésben rendelt szövegek
10. A KÉPERNYŐ 12
   10.1. Alfanumerikus képernyő
      10.1.1. Karakteres üzemmódok és beállításuk
      10.1.2. Kiírás képernyőre 13
      10.1.3. Speciális karakterek a képernyő-kép szerkesztéséhez
      10.1.4. Escape-szekvenciák 14
      10.1.5. Alfanumerikus képernyő használatával kapcsolatos címek
   10.2. Grafikus képernyő 15
      10.2.1. Grafikus üzemmódok
      10.2.2. Grafikus üzemmódok kiválasztása
      10.2.3. Grafikus/rajzoló utasítások 16
      10.2.4. Szövegkezelés grafikus módban
      10.2.5. SPRITE-ok
11. LEMEZ I/O (VC1541 egység) 17
   11.1. A lemezegység parancsai
      11.1.1. A BASIC 3.5 lemez-parancsai
      11.1.2. A BASIC 2.0 lemez-parancsszövegei 18
   11.2. Szekvenciális file-ok kezelése
   11.3. Random (közvetlen elérésű) file-ok kezelése 19
   11.4. Fizikai-blokk kezelő utasítások 20
   11.5. Relatív file-ok kezelése
   11.6. FloPPy-memória kezelése 21
12. NYOMTATÓ (VC1525 / MPS801 / MPS803 mátrixPrinter) KEZELÉSE 22
   12.1. Kiírás nyomtatón - I/O utasítások
   12.2. Nyomtatás-vezérlő karakterek 23
13. RS232 INTERFACE HASZNÁLATA 24
   13.1. I/O utasítások
   13.2. Az RS232 állapotregiszterben megjelenő státuszok 25
   13.3. RS232 memória-címek
14. KAZETTÁS SZALAGEGYSÉG KEZELÉSE 26
   14.1. A szalagos file-ok szerkezete
   14.2. I/O utasítások 27
   14.3. Speciális I/O-k (Program tárolása/betöltése)
   14.4. Fontosabb memóriacímek
15. BOTKORMÁNY 28
   15.1. A botkormány használata
   15.2. Felhasznált memóriarekeszek (TED regiszterek)
16. HANG-KIMENET KEZELÉSE 29
   16.1. Általános információk
   16.2. Az előállított hang jellemzői
   16.3. A hanggenerátorok (oszcillátorok) megszólaltatása 30
   16.4. Frekvenciaadatok 31

GRAFIKA

1. GRAFIKUS LEHETŐSÉGEK 32
1.1. A TED grafikus regisztereinek kiosztása
1.2. Választható üzemmódok 33
2. KARAKTERES ÜZEMMÓD
2.1. Standard mód 34
2.2. User mód
2.3. 256 karakteres mód 35
2.4. Többszín mód
2.5. Bővített háttérszín mód
3. NAGYFELBONTÁSÚ GRAFIKA 36
3.1. Standard mód
3.2. Többszín mód 37
4. EGYEB GRAFIKUS LEHETŐSÉGEK
4.1. SPRITE-ok (MOB-OK)
4.2. Scrolling (görgetés)
4.2.1. Karakter szintű scroll (karakteres képernyőn)
4.2.2. Finom (Pontszintű) scroll
4.3. Képernyő elsötétítése 38
4.4. TED leállítása 39
4.5. NTSC szabványú videojel előállítása
4.6. Színárnyalatok
4.7. Villogás
4.8. Speciális képernyő-kezelés 40

GÉPI-KÓDÚ PROGRAMOZÁS

1. GÉPI-KÓDÚ UTASÍTÁSOK 41
2. PÁRHUZAMOS TÁBLÁZATOK 43
3. A KERNAL 44
3.1. KERNAL táblázatok
3.2. KERNAL rutinok
3.3. KERNAL státuszok 46
4. GÉPI-KÓDÚ RUTINOK HASZNÁLATA BASIC PROGRAMOKBÓL 47
4.1. BASIC Programok kezelésével kapcsolatos memóriacímek
4.2. Szabadon felhasználható területek 48
4.3. A BASIC interpreter és a KERNAL főbb user-belépési pontjai 49
4.4. RAM és ROM olvasó rutinok 50
4.5. Rutinhívás és paraméterátadás BASIC Programból
4.5.1. A POKE és PEEK használata
4.5.2. BASIC szöveg kezelése gépi-kódú rutinokból 51
4.5.3. Lebegőpontos műveletek gépi-kódban
4.5.4. Példák a fenti rutinok felhasználására 52
5. A MEMÓRIA KEZELÉSE 53
6. KÜLSŐ CSATLAKOZÓK 54

A TEDMON MONITOR HASZNÁLATA 55

FÜGGELÉK

A. SZÍNKÓDOK 56
B. SZÍNKOMBINÁCIÓK
C. A LENYOMOTT BILLENTYŰ KÓDJA (LOC. 198 / $C6 TARTALMA)
D. ASCII KARAKTEREK(decimális táblázat) 57
E. HEXADECIMÁLIS-DECIMÁLIS KONVERZIÓ 58
F. KÉPERNYŐ (SCREEN) KÓDOK 59
G. BASIC KULCSSZAVAK, RÖVIDÍTÉSÜK ÉS TOKENJEIK 60
H. A MÁGNESLEMEZ MINT ADATHORDOZÓ 62
H.1. A katalógus (directory)
H.2. File-ok 63
I. A LEMEZEGYSÉG HIBASTÁTUSZAI 64
J. PROGRAMOZÁSI PÉLDÁK 65
J.1 Várakozás billentyű leütésére a WAIT utasítás felhasználásával
J.2. Példák a billentyűPuffer használatára
J.3. BASIC-gépi-kód interface (Pszeudoregiszterek használata)
J.4. A raszter-interrupt felhasználása 66
J.5. BASIC szöveg feldolgozása gépi-kódból
K. A ROM ÉS A RAM TÉRKÉPE 68
L. FELHASZNÁLT ÉS AJÁNLOTT IRODALOM A COMMODORE 16-HOZ 76

BEVEZETÉS

E zsebkönyv készítésekor - C64-hez készült elődjéhez hasonlóan - azt tűztük ki célul, hogy az egyébként igen sokat tudó C16 hatékony használatához szükséges minden lényeges információt összegyűjtsünk és sűrített, de mégis áttekinthető és könnyen kezelhető formába foglalva tegyük hozzáférhetővé.

Lehetőleg kerültük a hasonló jellegű könyveknél megszokott abc-rendben történő felsorolást, helyette a logikai sorrendű tematikus tárgyalást alkalmaztuk, olyan könnyen kezelhető egységekben, amelyek a hozzájuk tartozó adatokkal együtt is áttekinthetőek mindenki számára. A gyorsabb tájékozódás érdekében az éppen tárgyalt téma rövid és hosszabb címét a lapok jobb illetve bal felső sarkában külön is feltüntettük.

Referencia-könyv jellege ellenére ott, ahol úgy éreztük, hogy a száraz információ önmagában nem sokat mond, de frappáns és rövid példával jól érzékeltethetjük a dolog lényegét, kisebb kódrészleteket is bevettünk az anyagba. Sok esetben azonban be kellett érnünk azzal a megyőződéssel, hogy aki az adott információt használni akarja, az tudni fogja, mi az és mire használható.

JELÖLÉSEK ÉS KONVENCIÓK:

A szintaxis-leírásokban szereplő nagybetűs részek BASIC-kulcsszavak, vagy más okból kötött formájú szintaktikus elemek, azaz változtatás nélkül használandók. (Megjegyzendő, hogy a BASIC kulcsszavak nagy része begépelésnél ennek ellenére rövidíthető, lsd. G. FÜGGELÉK). A kisbetűvel szereplő részek jelölések, és értelemszerűen helyettesitendők be. Ott, ahol az alábbi egyszerű jelölések önmagában való alkalmazása a kötelező elválasztójelek hiánya vagy egyéb ok miatt értelemzavaró lenne, a jelöléseket (az értelemszerű behelyettesitéskor elhagyandó) < > jelek közé zárjuk.

Az alkalmazott rövidítések és jelek:

Szintaktikus elemek:

konst - konstans (állandó)
vált - változó
kif - kifejezés, azaz: konstans, változó vagy kifejezés
dummy - tetszőleges aktuális értékű változó vagy konstans (egyes függvények argumentumaként)
[elem] - opcionális szintaktikus elem (változó, kifejezés, stb.) (az alapértelmezésszerű értéket széles betűvel szedtük)
[elem<delim>...] - tetszőleges számban ismételhető elem (<delim> az elemek között alkalmazandó elválasztójel)
lista - [elem,...] (szintaktikus elemek vesszővel elválasztott sorozata

A fenti fogalmak módositására, ill. szükitésére módosító jelek előzhetik meg a fogalmat:

l - logikai, pl: lkif = logikai kifejezés, értéke TRUE/FALSE
b - byte, azaz 0-255 értékkészletű numerikus, egész
c - cím, azaz 0-65535 értékkészletű numerikus, egész

Bizonyos esetekben a módosítót az elemhez hátulról ragasztjuk, a BASIC-típusjelölésnek megfelelően:

% - egész, (pl: vált%) [-32768,32767] értékkészlettel
$ - füzér (szöveg,string), azaz pl: konst$ - "szöveg" (szöveg konstans idézőjelek között) v. kif$ - lehet szövegkonstansok és/vagy szövegváltozók "+" operátorral összefűzött sorozata, stb.
# - numerikus (szám)

Egyéb jelölések (megjegyzés rovatban)

o - opcionális, nem kötelező a használata
f - függvény
P - csak Program-módban használható
d - csak direkt-módban - sorszám nélkül - adható ki

1. SZINTAKTIKUS ELEMEK ÉS REPREZENTÁCIÓJUK:

1.1 Szintaktikus elemek

Konstansok:

e9észek: [-32768,32767] értékkészletű e9ész számok
lebe9ő Pontos számok:
decimális Pontot és/va9y E-t(exPonens részt) tartalmazó számok. (A fenti tartományon kívül eső e9észek is íg9 tárolódnak). Abszolút értékük 2.9387E-39 és 1.70141E+38 közé esik.
szöve9/karakter konstansok:
karaktersorozat idézőjelek közé zárva (max. 255 kar.)
lo9ikai: i9az = -1 hamis = 0

Változók:

Nevük első karaktere csak betű, a többi alfanumerikus karakter lehet. Ezt követheti az e9yes típusokra kötelező típusjel. A név első két karaktere és az esetle9es típusjel (%,$) azonosítja őket. A teljes hossz max. 255, és beá9yazva sem tartalmazhat BASIC alaPszót!
Mindhárom típusban me9en9edett skalár és tömbváltozók használata. Tömbváltozók esetében a változó nevét zárójelek között a DIM utasításban deklarált számú index követi, e9ymástól vesszővel elválasztva.

lebe9ő Pontos(valós) változók: nincs típusjel
e9ész változók típusjele: %
szöve9(strin9) és karaktervváltozók típusjele: $

Kifejezések:

Konstansok és/va9y változók és/va9y fü99vények között a típusnak me9felelő oPerátorokkal kijelölt művelettel, az utasítás vé9rehajtása során kiszámításra kerülő érték. A műveletek Prioritásuk sorrendjében hajtódnak vé9re, azonos szint esetén balról-jobbra.

OPerátorok (Prioritási sorrendben):

aritmetikai:	() ↑ * / + -
strin9:	+ (konkatenálás)
relációs:	= <> < > <= >= => =< (azonos szintűek)
lo9ikai:	NOT AND OR

Utasítások:

Kötött formájú karaktersorozatok (kulcsszavak), előírt Paraméterekkel/oPerandusokkal kie9észítve, amelyek va9y a Pro9ram vé9rehajtását, va9y az oPerációs rendszer tevékenysé9ét írják elő, illetve vezérlik.

Utasítássor:

Sorszámmal kezdődő utasítássorozat, max. 80 karakter hosszban. Az utasításokat kettősPonttal kell elválasztani. A kulcsszavakban szóköz nem lehet, de a kulcsszavak, nevek és oPerátorok közötti szóközök szüksé9telenek. A kulcsszavak rövidítése me9en9edett, lsd. G.FÜGGELÉK

Parancssor:

Sorszám nélkül, be9éPelt utasítás, amelyet a RETURN beütésekor az interPreter azonnal vA9rehajt. E9yes BASIC-elemek (lsd. ott) csak Pro9ram-módban használhatók, mások (Pl. MONITOR,NEW) mellékhatásuk miatt főle9 Parancs-módban használatosak.

1.2. Az ismertetett elemek memóriabeli rePrezentációja:

konstansok :

e9észek : két byte (hi9h,low)
valósak : 5 byte
szöve9/karakter : 3byte+1byte/karakter

változók :

```
e9észek  : 2(név)+2(érték:Hi,Lo)+3(üres)          byte
valósak  : 2(név)+1(exPonens)+4(mantissza)        byte
szöve9   : 2(név)+1(hossz)+2(Pointer)+2(üres)     byte
fü99vény : 2(név)+2(def.Ptr)+2(vált.Ptr)+1(üres) byte

Pointer(Ptr): belső felhasználású cím (nincs BASIC me9fele-
              lője). A 9éPi-kódú Pro9ramozás konvenciói szerint
              a teljes 64 Kb-os címtartományt leíró 2 byte közül
              az első az alacsonyabb (low,lo), a második a ma9a-
              sabb helyértékű (hi9h,hi) biteket tartalmazza

tömbök   : 5byte(név)+2byte*dimenziószám+
           +elemszám* 2byte (ha inte9er)
                      5byte (ha valós)
                      3byte+1byte/karakter (ha szöve9)
```

BASIC-kulcsszavak és oPerátorok:

1 byte-os kódok (token-ek)

utasítássor:

```
2 byte (link address lobyte/hibyte) : köv. sor címe
2 byte (sorszám lobyte/hibyte) : bináris szám (1-63999)
n byte (max.79) BASIC-szöve9 (tokenek, szóközök, stb)
1 byte (0) sor-vé9e jel
```

Pro9ram-vé9e:

3 byte (sor-vé9e:0 + link-address: 00)

A fentiek alaPján a BASIC Pro9ram mérete csökkentésének eszközei, szüksé9 esetén:

- több utasítás írása e9y sorba (kevesebb sorszámra van szüksé9)
- BASIC kulcsszavak rövidítése (fenti cél érdekében)
- rövid változónevek használata
- többször használt konstans helyettesítése változóval
- konstans-táblázatok DATA-ként való tárolása
- ahol lehet, e9ész-típusú tömbök használata
- TAB, SPC használata a kurzor Pozicionálására
- szóközök elha9yása (csak vé9szüksé9 esetén, de 9yorsít is)
- REM utasítások törlése (csak vé9szüksé9 esetén, de 9yorsít is)
- overlay használata (lásd J.2)

1.3. A BASIC-Pro9ramok tárolásával kaPcsolatos főbb memóriacímek:

(lásd mé9 GEPIKóD-fejezet, 4. Pont)

BASIC Pro9ram kezdőcíme (Pointer)	TXTTAB	43,44	$2b,$2c
BASIC által használt le9ma9asabb cím	MEMSIZ	55,56	$37,$38
Kürrens BASIC-sor sorszáma	CURLIN	57,58	$39,$3a
Előző BASIC-sor sorszáma	OLDLIN	601,602	$259,$25a
CONT visszalépési sorszám (Ptr)	OLDTXT	603,604	$25b,$25c
BASIC változók kezdőcíme (Pointer)	VARTAB	45,46	$2d,$2e
Kurrens BASIC-változó neve	VARNAM	69,70	$45,$46
Kurrens BASIC-változó adat címe (Ptr)	VARPNT	71,72	$47,$48
BASIC tömbök kezdőcíme (Pointer)	ARYTAB	47,48	$2f,$30
Szabad memória kezdőcíme (Pointer)	STREND	49,50	$31,$32
Strin9-tárolóterület alja (Pointer)	FRETOP	51,52	$33,$34
Strin9-Pointer		80-83	$50-$53
Kurrens DATA-sor sorszáma	DATLIN	63,64	$3f,$40
Kurrens DATA-elem címe (Ptr)	DATPTR	65,66	$41,$42
Hibás sor sorszáma		1264,1265	$4f0,$4f1
Hibakód		1263	$4ef
Hiba címe		1269,1270	$4f5,$4f6
TRAP-sor sorszáma		1266,1267	$4f2,$4f3
FOR/NEXT ciklusváltozó (Ptr)	FORPNT	73,74	$49,$4a
kurrens DO címe	DOPNT	1272,1273	$4f8,$4f9
DO-sor sorszáma		1274,1275	$4fa,$4fb
REPEAT-számláló		1345,1346	$541,$542
INPUT - Pointer		67,68	$43,$44
.A re9iszter tárolóhely (SYS-hez)	SAREG	2034	$7f2
.X re9iszter tárolóhely (SYS-hez)	SXREG	2035	$7f3
.Y re9iszter tárolóhely (SYS-hez)	SYREG	2036	$7f4
.ST re9iszter tárolóhely (SYS-hez)	STREG	2037	$7f5

2. A BASIC 3.5 ALAPUTASíTÁSAI:

2.1. Deklaratív utasítások:

A BASIC egyszerű változók és max. 11 elemű egydimenziós tömbök deklarációját nem igényli. Az interPreter a változó Programvégrehajtás közbeni első előfordulásakor foglalja le a szükséges memóriát, rögzíti a változó attributumait és inicializálja.

tömbök	DIM [név(kif#-lista),...]	P
függvények	DEF FN név(változó#)=kif#	P
konstansok	DATA érték-lista	P

2.2. Értékadó utasítások:

változónak	[LET] változó=kifejezés
DATA-listából	READ változó-lista
memóriarekesznek	POKE ckifejezés,bkifejezés

2.3. Programszervező utasítások:

ugró utasítások:

feltétlen	GOTO sorszám
feltételes	IF lkifejezés THEN sorszám
	IF lkifejezés GOTO sorszám
kiszámított	ON bkifejezés GOTO sorszám-lista

feltételes utasítás IF lkif THEN [utasítás:...]
[:ELSE [utasítás:...]]

ciklusszervezés:

I.véges ciklus

ciklus kezdése	FOR ciklusvált#=kif# TO kif# [STEP kif#]
ciklus zárása	NEXT [ciklusváltozó#]

II.végtelen ciklus (kilépéssel)

ciklus kezdése:	
kilépés, ha lkif=igaz	DO [UNTIL lkif]
kilépés, ha lkif=hamis	DO WHILE lkif
feltétel nélküli kiugrás a ciklusmagból a következő utasításra	EXIT
ciklus zárása:	
kilépés, ha lkif=igaz	LOOP [UNTIL lkif]
kilépés, ha lkif=hamis	LOOP WHILE lkif

szubrutinhívás (max 23. mélységig):

feltétlen	GOSUB sorszám
kiszámított	ON bkif GOSUB sorszám-lista
gépi kódú rutiné	SYS ckifejezés

visszatérés

BASIC rutinból	RETURN
saját hibakezelő rutinból(TRAP)	RESUME

Program vége: END

futás megszakítása STOP

megjegyzés REM megjegyzés-szöveg

3. FÜGGVÉNYEK

A matematikában megszokott függvényfogalomnak megfelelően a kifejezésekben mindenütt helyettesíthetik az eredményükkel azonos típusú konstansokat, vagy változókat. Pillanatnyi értékük az argumentumuknak megfelelő behelyettesítési érték.

Funkció	Szintaxis		Eredmény típusa	Megj
3.1. Matematikai függvények				
abszolút-érték	ABS(kif#)		#	
egészrész	INT(kif#)		#	
előjel	SGN(kif#)		#	
ex	EXP(kif#)		#	
ln(x)	LOG(kif#)		#	
x^0.5	SQR(kif#)		#	
arctg(x)	ATN(kif#)		#	
cos(x)	COS(kif#)		#	
sin(x)	SIN(kif#)		#	
tg(x)	TAN(kif#)		#	
random (Pszeudo)	RND(kif#)	ha kif#>0	#	
(random)		ha kif#=0		
(Pszeudo,reset)		ha kif#<0		
user által DEF FN - nel definiált függvény	FN név(kif#)		#	
user által definiált gépi kódú függvény	USR(kif#)		#	
3.2. Karakter és stringkezelő függvények				
megfelelő ASCII karakter	CHR$(kif#)		$	
első kar.numerikus értéke	ASC(kif$)		b	
nkif. decimális szám-string9gé konvertálása	STR$(kif#)		$	
nkif. 4-digites hexa-string9gé konvertálása	HEX$(ckif)		$	
max.4 karakteres hexa-string9 decimális számmá konvertálása	DEC(hexa-digit$)		c	
string9 baloldala (n# kar)	LEFT$(kif$,n#)		$	
string9 közepe (kezd#-től)	MID$(kif$,kezd#,n#)		$	
string9jobboldala (n# kar)	RIGHT$(kif$,n#)		$	
string9 hossza	LEN(kif$)		b	
numerikus string9 értéke	VAL(kif$)		#	
részstring9 kezdőpozíciója a string9 első/kezd#-edik karakterétől kezdve	INSTR(kif$,rész$,[kezd#])		b	
3.3. Memóriakezelő függvények:				
szabad memória mérete	FRE(dummy)		%	
adott memóriacím tartalma	PEEK(ckif)		b	
3.4. Grafikus függvények:				
kurzor pozíciója a sorban	POS(dummy)		b	
Pixel-kurzor koordinátája ha kif#=0, x-koordináta kif#=1, y-koordináta kif#=3, szín-forrás	RDOT(kif#)		%	
kif#-kódú(0-4) színforrás színe (1-16)	RCLR(kif#)		%	
kif#-kódú(0-4) színforrás fényerő-szintje	RLUM(kif#)		%	
grafikus üzemmód kódja	RGR(dummy)		%	
3.5. Egyéb függvények				
botkormány pillanatnyi állása (lsd. ott)	JOY(kif#)		%	
hibakódnak megfelelő hiba-üzenet szövege	ERR$(bkif)		$	

4. RENDSZERVÁLTOZÓK

bekapcsolás óta eltelt idő (1/60-ad sec)	TI	%
eltelt idő (string-formában)	TI$	$
utolsó I/O végrehajtásának státusza	ST	b
utolsó hiba száma	ER	b
hibás BASIC sor sorszáma	EL	c
lemezmeghajtó státusza	DS	b
lemezmeghajtó hibaüzenete	DS$	$

5. GRAFIKUS ALKALMAZÁSOKAT SEGÍTŐ UTASÍTÁSOK:

Részletesebben lásd még 15-16 old. (VIDEO 10.2. pont)

grafikus képernyő derékszögű részletének szövegváltozóba való tárolása (max. 255 byte kép)	SSHAPE vált$,x1,y1[,x2,y2]
szövegváltozóba tárolt kép megjelenítése a képernyő megadott pozíciójában, a kívánt transzformációval	GSHAPE vált$[,x1,y1][,transzf]
grafikus képernyőpontjaihoz tartozó koordináták skálafaktorának megváltoztatása a [0..319,0..199]-ről [0..1023,0..1023]-ra	SCALE 1
skálafaktor visszaváltoztatása normál értékre	SCALE 0

6. RENDSZER FUNKCIÓK

dinamikus várakoztatás egyenértékű: 10 IF (PEEK(cím) XOR bmask2) AND bmask1)=0 GOTO 10 (ld. APPENDIX J.1)	WAIT cím,bmask1[,bmask2]	
dinamikus memóriafoglalás felszabadítása	CLR	
DATA-pointer alaphelyzetbe állítása (memória file rewind-ja)	RESTORE	
program és változók törlése	NEW	
program indítása első vagy megadott sortól	RUN [sorszám]	
félbeszakított programfutás folytatása (csak akkor lehetséges, ha nem hiba okozta a leállást, hanem END, STOP vagy RUN/STOP gomb lenyomása)	CONT	d

7. PROGRAMOZÁSI SEGÉDESZKÖZÖK

automatikus program-sorszámozás	AUTO növekmény	d
AUTO parancs hatásának megszüntetése	AUTO	d
program újra-sorszámozása (hivatkozásoké is pl. GOTO, GOSUB stb)	RENUMBER [u.sor#],[növekmény],[régikezdősor#]	d
programsorok törlése	DELETE [kezdősor#][-végsorszám]	d
program megadott sorainak listázása	LIST [kezdősorszám][-végsorszám]	
funkcióbillentyűk hozzárendelése (összesen max. 128 karakter)	KEY billentyű#,szoveg$	
funkcióbillentyűk hozzárendelésének listázása	KEY	
BASIC-nyomkövetés bekapcs.	TRON	
BASIC-nyomkövetés kikapcs.	TROFF	
MONITOR-ba való átlépés	MONITOR	
saját hibakezelő rutin aktiválása (hiba esetén a vezérlés ráadódik)	TRAP sorszám	
saját hibakezelés felfüggesztése	TRAP	

8. PERIFÉRIÁK KEZELÉSE, ADAT BE ÉS KIVITEL

8.1. Speciális I/O utasítások Programok be és kivitelére

Program-módban való használatuk különös gondosságot igényel, mellékhatásaik miatt (Program felulírása, file-ok lezárása stb.)

Program (sorszámozott memória-file) listázása képernyőre	LIST [kezdősorszám][-végsorszám]
Program betoltése	LOAD Programnév$,[egység#][,típ#] ahol típ#: 0 - a Program a BASIC terület elejére töltődik 1 - a kimentéskor elfoglalt helyére töltődik egység#: 1 - szalag (elhagyható) 8/9/10/11 - lemez
Program kimentése	SAVE Programnév$,[egység#][,típ#] ahol: típ#: 0/1 - szalag esetében a címketípus BASIC/gépikód lesz
Program ujrakimentése (már létező file-ra alatt)	SAVE "@[meghajtó]:"+Programnév$, egység#[,tip#] ahol meghajtó: duál-egységnél (0/1) pl.: SAVE "@0:TABLOZO",8
kimentés sikeres voltának ellenőrzése	VERIFY Programnév$,egység#

A BASIC 3.5 mágneslemez-egységre (disk) a fenti utasítások speciális, rovidebb formáit is megengedi

Program betoltése	DLOAD Pronév$[,Dmeghajtó][,Uegység#] ahol egység#: 8(alapért.) v. 9/10/11 meghajtó 0(alapért.) vagy 1
Program kimentése	DSAVE Pronév$[,Dmeghajtó][,Uegység#] ahol egység#: 8/9/10,11 meghajtó : 0/1
lemez tartalomjegyzékének, vagy csak egyes Program(ok) katalógusbejegyzésének listázása	DIRECTORY [Dmeghajtó],[Uegység#], [Pronév$]

8.2. Az utasítások általános formája

♦A képernyőn és a billentyűzeten kívül (rendszer egységek) minden más Periférián végrehajtandó adat be- és kivitel (I/O) előtt az alábbi lépések végrehajtása szükséges
- fizikai és logikai egység egymáshoz rendelése
- logikai egység megnyitása, attributumok megadása
- üzemmód beállítása

E funkciók végrehajtására az OPEN, a vele már megnyitott file attributumainak módosítására pedig a PRINT# utasítás használható.

♦Ezután következhetnek az I/O utasítások
- adatkivitelre PRINT (képernyőre) ill. PRINT# (más egységre)
- adatbeolvasásra INPUT, GET ill INPUT#, GET#

Lehetőség van az eredetileg képernyőre szánt (LIST v. PRINT utasítással készülő) output más Perifériára való irányítására a CMD utasítással.

♦Az adatátvitel végrehajtása után a külső Perifériákra megnyitott file-okat le kell zárni. erre szolgál a CLOSE utasítás.

file megnyitása

(Egyszerre legfeljebb 10 file lehet nyitva)

```
OPEN file#,egység#,csatorna#,szöveg$
  ahol:
    file#: logikai fileazonosító, értéke
      1-255, ezen belül:
        1-127 : normál azonosító
        > 127 : minden CR után egy LF-
          karakter is kiadódik (nem-
          standard nyomtatókhoz)

    egység#  0 - billentyűzet
             1 - szalagegység
             2 - RS232 csatorna
             3 - képernyő
             4,5 - nyomtató
             8,9,10,11 - lemez

    csatorna#: (másodlagos cím)
       0 - szalagos READ
           v. "LOAD" rendszer-csatorna
       1 - szalagos WRITE
           v. "SAVE" csatorna
       2 - mint 1, de file végén EOT
           (szalagvége) címke íródik
       2-14 adat-csatornák
       15 - Parancs csatorna(lemeznél)

    szöveg$: adat vagy Parancs-szöveg,
       szerkezete készülék-típusonként
       különböző.
```

Adat vagy Parancs kiküldése:

```
PRINT [kif<ctrl>...]     (képernyőre)
PRINT#file#,[Parancs$][,kif<ctrl>...]
   ahol:
     file#: OPEN-ben hozzárendelt logikai
            file-azonosító
     Parancs$: készülék és file-típus
            függő
     <ctrl> file-típustól függő vezérlő-
            vagy elválasztójel
     kif: kiírandó érték/adat
```

Megj: függetlenül a kiírandó adat-elemek között alkalmazott elválasztó-jelektől (vessző, Pontosvessző stb. a PRINT# is csak legfeljebb szóköz-jelet tesz elválasztó jelként az outPut adatelemei közé, ami a file visszaolvasásakor a lista elemeinek szétválasztását nem teszi lehetővé, ezért az elemek kiírásakor a visszaolvasáskor használandó elválasztójelet (",", chr$(13)) külön, adatként kell kiírni.

Karakterenkénti beolvasás (változónként egy karakter)

```
GET változó$-lista     (billentyűzetről)
GET#file#,változó$-lista
   ahol:
     file#: mint fent.
```

Adatbeolvasás típusellenőrzéssel RETURN-ig (chr$(13)-ig).
(Nem az utolsó változót követő delimiterig, hanem chr$(13)-ig olvas, a felesleges beolvasott byte-ok / adatelemek elvesznek)

```
INPUT [konst$;]változó-lista     (bill.)
INPUT#file#,változó-lista
   ahol:
     file#: mint fent
     változó-lista: egy-egy változóba
        két delimiter-jel között levő
        információ kerül bele, típus-
        ellenőrzéssel. Delimiter lehet
        RETURN vagy vessző. Kettős-
        Pont nem lehet delimiter.
```

File lezárása

```
CLOSE file#
   ahol:
     file#: mint fent
```

8.3. Az I/O utasítások által használt legfontosabb címek

aktuális file-név/Parancs kezdőcím-Ptr	FNADR	175,176	$af,$b0
aktuális file-név hossza	FNLEN	171	$ab
aktuális logikai file#	LA	172	$ac
aktuális csatorna# (másodlagos cím) a csatorna állapotától függő érték hozzá	SA	173	$ad
aktuális egység#	FA	174	$ae

9. BILLENTYŰZET

A billentyűzeten beütött adatok egy max. 10 karakteres FIFO-puffetbe kerülnek felhasználásukig (INPUT, GET), ahonnan a GET utasítással karakterenként is kiolvashatók. A puffetbe másként (pl. POKE-kal) beírt karakterekkel szimulálni is lehet a billentyűzet használatát, ami érdekes felhasználásokra ad lehetőséget (pl. Program-szegmentálás, önmódosító-Program, lsd FÜGGELÉK J.2).

9.1. Beolvasás billentyűzetről

várakozással, típusellenőrzéssel	INPUT változó-lista
u.a. kérdéskiírással	INPUT "kérdés";változó-lista
billentyű-pufferből, várakozás nélkül (1 jel/változó)	GET változó$-lista
u.a. várakozással billentyűleütésre	GETKEY változó$-lista

9.2. A billentyűzet működésével kapcsolatos főbb memóriahelyek:

tartalom	cím dec	cím hexa
Puffer-tárolóhely (10 karakter hosszúságú)	1319-1328	$527-$530
Puffer mérete	1343	$53f
utóljára lenyomott billentyű (64 = egy sem)	198	$c6
előzőleg leutott billentyű	2038	$7f6
a Pufferban levő karakterek száma	239	$ef
karakterismétlés engedélyezése($80)	1344	$540
ismétlés gyakorisága	1345,1346	$541,$542
betűváltó jelzése	1347	$543
betűkészlet-váltás (C= + SHIFT)	1351	$547
engedélyezése ($00) vagy letiltása ($80)		
billentyűzet-mátrix dekódoló-tár címe(Ptr)	237,238	$ec,$ed
billentyűzet-táblázat Pointer	1349,1350	$545,$546
aktuális billentyű mátrix kód	238	$ee
idézőjel mód flag	203	$cb
billentyűzet/képernyő flag	208	$d0
INSert mód flag	207	$cf
STOP flag	145	$91
kurrens funkció-gomb indexe	1374	$55e
szöveghossza	1373	$55d
funkció-gombok szöveghossz-táblázata	1375-1382	$55f-$566
szövegtáblázata	1383-1511	$567-$5e7
színbillentyű-kód RAM/ROM flag	2041	$7f9
INPUT CR-flag	199	$c7
sorvég INPUT-nál	195	$c3

9.3. A billentyű-mátrix (oszlop-kiválasztó regiszter) 64816 $fd30
(mátrix-olvasó regiszter) 65288 $ff08

oszlop-kiválasztó reg. bitjei

	0	1	2	3	4	5	6	7
0	del	3	5	7	9	cur	cup	1
$ 1	retn	W	R	Y	I	P	* home	cur/cul: jobb/bal
↑ 2	£	A	D	G	J	L	]; ctrl	shft:shift
↑ 3	help	4	6	8	0	cul	cud 2	esc:escape
0 4	f1	Z	C	B	M	>.	esc spc	spc:szóköz
^ 5	f2	S	F	H	K	[:	= comm	comm:commodore
6	f3	E	T	U	O	~	+ Q	retn:return
bit 7	@ shft	X	V	N	<,	?/ stop		

Használata: A $fd30 regiszterben 0-ra kell állítani az ellenőrizni kívánt mátrix-oszlopnak megfelelő bitet (a többit 1-re). A TED a $ff08 regiszterbe való írás pillanatában olvassa a billentyű-mátrix megfelelő oszlopát, és az eredményt a $ff08 regiszterbe teszi. Innen kiolvashatjuk hogy az adott oszlop mely billentyűi vannak lenyomva. Ha szükséges, ezután áttérhetünk a következő mátrix-oszlop ellenőrzésére (miután az interrupt-rutin ezt minden meghívása alkalmával elvégzi, célszerűen gépi kódú rutinban, az interrupt letiltása mellett), lsd. GÉPIKÓD-rész, 4.5.1. fejezetét.

9.4. A funkcióbillentyűkhöz alapértelmezésben rendelt szövegek:

```
F1:  "GRAPHIC"                    F4  "SCNCLR"+CHR$(13)
F2:  "DLOAD"+CHR$(34)             F5: "DSAVE"+CHR$(34)
F3   "DIRECTORY"+CHR$(13)         F6: "RUN"+CHR$(13)
HELP: "HELP"+CHR$(13)             F7: "LIST"+CHR$(13)
```

10. A KEPERNYŐ

A képernyőnek 5 BASIC-ből beállítható üzemmódja van:

0 - alfanumerikus (25 sor * 40 oszlop) - alapértelmezés
1 - nagy felbontóképességű grafikus (HI-RES, 320 * 200 pont)
2 - osztott (20 sornyi HI-RES + 5 sor alfanumerikus)
3 - többszínű grafikus (MULTI, 160 * 200 pont)
4 - többszínű grafikus + szöveg (20 sor MULTI + 5 sor alfanum.)

A képernyőn alapvetően három mezőt különböztetünk meg:

-a képernyő keret
-a képernyő alap (kereten belüli rész, háttér)
-a kurzor (karakterek, előtér)

E mezők színét egy-egy regiszter tárolja, az előtér adott pozíciójának színe ezenkívül még az ún. szín-mátrixba is bekerül, a képernyő-kép generálásához onnan veszi a TED. Ezért az előtér színe minden karakterpozícióban különböző lehet, míg a keret mindig, a háttér színe (továbbá MULTI módban a további két szín) általában egyféle (POKE utasításokkal többszínű háttér is beállítható, lsd. GRAFIKA c. fejezet, 2.5 pont).

Az üzemmód beállítása	GRAPHIC üzemmód[,törlés] ahol: üzemmód: 0-4 (lsd. fenn) törlés: Ø - a képernyő nem törlődik 1 - a képernyő törlődik is	
Színregiszter (és az általa jellemzett képernyő-mező) színének beállítása	COLOR sreg,szín[,fényesség] ahol sreg: 0 - háttér 1 - előtér (kurzor) 2 - többszínű grafikus 1 3 - többszínű grafikus 2 4 - keret szín: 1-16, lsd. APPENDIX A. fényesség: Ø-7 (min-max)	
Képernyő törlése bármely üzemmódban	SCNCLR	
Aktuális grafikus üzemmód kódjának lekérdezése (függvény)	RGR(dummy)	%
Valamely színregiszter pillanatnyi színének lekérdezése (függvény)	RCLR(sreg)	%
Adott színregiszterhez rendelt fényerő lekérdezése (függvény)	RLUM(sreg)	%

10.1. Alfanumerikus képernyő

A C16 képernyőkezelése a fentiekben ismertetett üzemmódokon túlmenően még ún. ABLAK-kezelésre és a képernyő görgetésére (scroll) is képes. Ezeknek a funkcióknak a karakteres képernyőn való kezelése speciális karaktersorozatok, ún. escape-szekvenciák képernyőre küldésével (lsd. alább) történhet.
Ha egy adott pillanatban létezik az ABLAK, az összes képernyőre menő információ az ABLAK-ban jelenik meg, illetve a parancsok csak azt érintik (kivéve az üzemmód és képernyőszín-beállító parancsokat).

10.1.1. Karakteres üzemmódok és beállításuk:

Alapértelmezés szerinti (karakteres) üzemmód (25sor * 40oszlop)	GRAPHIC 0[,törlés]
Osztott képernyő (HI-RES + 5 sor karakteres ill. MULTI+karakteres)	GRAPHIC 2[,törlés] GRAPHIC 4[,törlés]

10.1.2. Kiírás képernyőre

Alapértelmezés szerinti formátumban, tabulációval

PRINT nyomtatási-lista
ahol
nyomtatási-lista:
kif[<tab>kif<tab>...]
<tab>: ; - tabulálás nélkül
, - tabulálás (standard)
SPC(kif#) - kif# darab szóköz beszúrása
TAB(kif#) - pozicionálás kif#-edik oszlopba

Kiírás formátum megadásával

PRINT USING formátum$;nyomtatási-lista
ahol:
nyomtatási-lista: mint fenn
formátum$: a nyomtatási képet definiáló maszk-karakterlánc.

Lehetséges maszk-jelek:
: betű vagy szám
+ : szám, + előjel vagy szóköz
- : szám, - előjel vagy szóköz
↑↑↑↑ : tudományos jelölésmód exponens (E+##) rész
= : szöveg mezőközépre igazítása
> : szöveg jobbra-igazítása

Beszúrás-karakterek(a kiíratandó információnak eredetileg nem része, de a mezőben nyomtatásra kerül)
. : tizedespont helye a számban
, : vessző beszúrása számba
$: dollárjel a szám elé

(Meg): numerikus mező túlcsordulása esetén a mező %-gal lesz feltöltve, alfabetikus mező pedig mindig jobbról csonkul, ill. szóközzel jobbról töltődik fel)

Formátumos kiírásban 'beszúrás--karakterek' lecserélése más karakterekre

PUDEF "<blank><vessző><Pont><dollár>"
ahol:
<blank>: szóközt helyettesítő jel
<vessző>: vessző helyettesítő jel
<Pont>: Pont helyettesítő jel
<dollár>: dollárt helyettesítő jel

10.1.3. Speciális karakterek a képernyő-kép szerkesztéséhez:

FUNKCIÓ	röv.	billentyű	CHR$	hex	idézőjel-módban karakter (inverz)	hex
képernyő-törlése	(clr)	CLR	147	93		d3
kurzor alaphelyzetbe	(hom)	HOME	19	13		93
kurzor jobbra	(cur)		29	1d		9d
kurzor balra	(cul)		157	9d		dd
kurzor le	(cud)		17	11		91
kurzor fel	(cup)		145	91		d1
inverz karakter on	(rvs)	CTRL+9	18	12		92
inverz karakter off	(rvo)	CTRL+0	146	92		d2
villogás ON	(flo)	CTRL+,	130	82		c2
villogás OFF	(flf)	CTRL+.	132	84		c4
nagybetű-állás	(ucs)	SHIFT	142	8e		
kisbetű-állás	(lcs)		14	20		
RETURN	(rtn)	RETURN	13	0d		
DELETE	(del)	DEL	20	14		94
INSERT	(ins)	INST	148	94		d4
fekete	(blk)	CTRL+1	144	90		d0
fehér	(wht)	CTRL+2	5	5		85
vörös	(red)	CTRL+3	28	1c		9c
cián	(cyn)	CTRL+4	159	9f		df
bíbor	(pur)	CTRL+5	156	9c		dc
zöld	(grn)	CTRL+6	30	1e		9e
kék	(blu)	CTRL+7	31	1f		9f
sárga	(yel)	CTRL+8	158	9e		de
narancs	(org)	C= + 1	129	81		c1
barna	(brn)	C= + 2	149	95		d5
sárgászöld	(ygn)	C= + 3	150	96		d6
világospiros	(lrd)	C= + 4	151	97		d7
kékeszöld	(bgn)	C= + 5	152	98		d8
világoskék	(lbl)	C= + 6	153	99		d9
sötétkék	(dbl)	C= + 7	154	9a		da
világoszöld	(lgn)	C= + 8	155	9b		db

10.1.4. Escape-szekvenciák

Az alábbi funkciókat nem 1-1 karakter, hanem egy karakterpár, az ESCAPE -CHR$(27)- és egy, azt közvetlenül követő (alább ismertetendő) karakter valósítja meg. A hatás szempontjából közömbös, hogy a karaktersorozat billentyűzetről, vagy PRINT utasításból kerül a képernyőre.

képernyőablak bal felső sarkának beállítása a kurzor aktuális helyére	T	CHR$(84)
képernyőablak jobb alsó sarkának beállítása a kurzor aktuális helyére	B	CHR$(66)
képernyőgörgetés engedélyezése	L	CHR$(76)
képernyőgörgetés felfelé	V	CHR$(86)
képernyőgörgetés lefelé	W	CHR$(87)
képernyőgörgetés letiltása	M	CHR$(77)
képernyőméret csökkentése 23 sorra 38 oszlopra	R	CHR$(82)
képernyőméret visszaállítása	N	CHR$(78)
automatikus, folyamatos beszúrás (insert) mód	A	CHR$(65)
automatikus beszúrás megszakítása	C	CHR$(67)
egy sor betoldása	I	CHR$(73)
aktuális sor törlése	D	CHR$(68)
törlés sor elejétől a kurzorig	P	CHR$(80)
törlés kurzortól sor végéig	Q	CHR$(81)
ugrás aktuális sor elejére	J	CHR$(74)
ugrás aktuális sor végére	K	CHR$(75)
előző escape-funkció törlése	X	CHR$(88)
betoldás, idézőjel, inverz és villogó üzemmód törlése	O	CHR$(79)

ablak törlése: a CLR gomb kétszeri lenyomásával

10.1.5. Alfanumerikus képernyő használatával kapcsolatos címek

tartalom	cím dec	cím hex
képernyő-terület (karakteres mód)	2048- 4095	$0800-0fff
képernyő-cím felső-byte	1342	$53e
flagek		
grafikus-terület flag	117	$75
grafikus mód kód	131	$83
PRINT-mód flag	139	$8b
RVS ON flag	194	$c2
INSert flag	207	$cf
billentyűzet/képernyő flag	208	$d0
FLASH ON flag	1340	$53c
scroll-engedélyezés flag	2025	$7e9
auto-insert flag	2026	$7ea
EDITOR '""'-módban flag	203	$cb
PAL/NTSC flag	2045	$7fd
színek		
utolsó grafikus utasítás színregisztere (0-4)	132	$84
multicolor szín#1	133	$85
karakter színe	134	$86
karakter színe	1339	$53b
kurzor alatti szín	2029	$7ed
kurzor alatti jel SZÍN-mátrixbeli párjának címe	234-235	$ea-eb
kurzor		
kurzor sora	205	$cd
kurzor oszlopa	202	$ca
kurzor címe (kép-mátrixban)	200,201	$c8,c9
PIXEL-kurzor (PC) koordináták	685-688	$2ad-2b0
TAB-pozíció	9	$09
INPUT sor/oszlop	196-197	$c4-c5
előző karakter	2027	$7eb
ABLAK alsó sora	2021	$7e5
felső sora	2022	$7e6
bal szélső oszlopa	2023	$7e7
jobb szélső oszlopa	2024	$7e8
folytatósor bittáblázat	2030-2032	$7ee-7f0
VIDEO-mátrix címe (cím/256)		$7fb

10.... Grafikai képernyő

A C16 grafikus lehetőségei egyes vonatkozásokban felülmúlják a C64 lehetőségeit. Pl. osztott képernyő (grafikus és karakteres üzemmód egyidejű használata) egyszerűen, BASIC-ből megvalósítható; színenként 8-féle árnyalat érhető el; más tekintetben viszont alatta marad. Pl. nincsenek a hardver által megvalósított ún. 'sprite'-ok (egyes fordításokban 'foltok','szellemek', stb.), csak szoftver úton szimulálhatók. Az átlag-felhasználó szempontjából lényegesen fontosabb, hogy a grafikus lehetőségek nagyrésze BASIC nyelvből hozzáférhető és hatékonyan használható.

Nagyfelbontású (HI-RES) grafikus üzemmódban a képernyő minden (320*200) pontja, többszínű (MULTI) üzemmódban pedig pontpárjai (160*200) címezhetők. A címezhető pont neve PIXEL, az aktuálisan címzett pontra mutató pointer neve PIXEL-KURZOR (PC). Sok grafikus utasításban a PC által reprezentált PIXEL a rajzolási művelet alapértelmezésszerű kezdőpontja.

A rajzoló utasításokban x,y (abszolút) koordináták helyett mindenütt használhatunk elmozdulást is, amely mindig a PC pillanatnyi helyétől számítódik:

- előjeles számok/változók (x,y irányú eltolás) formájában
- polárkoordináta-szerűen (hossz és irányszög (fok) megadásával)

A szöget (mely sosem relatív) a felfelé irányuló függőlegestől az óramutató járásával megegyező irányban számítjuk.

10.2.1. Grafikus üzemmódok

A BASIC 3.5 grafikus-üzemmódban az alábbi lehetőségeket támogatja:

- nagyfelbontású (HI-RES)
- többszínű (MULTI) - 3 + 1 karakterpozíciónként választható szín
- SPRITE-ok (szoftver)
  vegyes (grafikus és karakteres) üzemmód
- osztott képernyő (grafikus képernyő + 5 soros szöveg-ablak)
- 16 alapszín (1-16),lsd. APPENDIX A.
- 8 színárnyalat (0-7)

A grafikus képernyő felépítése

- 200 * 320 képpont HI-RES módban
- 200 * 160 képpont MULTI módban
- 160 * 320 képpont (HI-RES) grafikus rész + 5 sor szöveg-ablak
- 160 * 160 képpont (MULTI) grafikus rész + 5 sor szöveg-ablak

10.2.2. Grafikus üzemmódok kiválasztása

Üzemmód beállítása	GRAPHIC üzemmód[,törlés] ahol: üzemmód: 1 - HI-RES 2 - HI-RES + szöveg 3 - MULTI 4 - MULTI + szöveg törlés: 0 - képernyőt nem törli 1 - képernyő törlése
Színregiszter (és az általa jellemzett képernyő - mező) színének beállítása	COLOR sreg,szín[,fényerő] ahol: sreg: 0 - háttér 1 - előtér, kurzor 2 - többszínű-1 3 - többszínű-2 4 - keret szín: 1-16 fényerő: 0-7 (min-max)
Képernyő törlése (bármely üzemmódban)	SCNCLR
Bit-térkép címzésének megváltoztatása a normál osztású (320*200) koordinátákról 1024*1024 osztásúra,vagy vissza	SCALE mód ahol: mód: 0 - normál mód 1 - 1024*1024
A 10 kbyte-os képernyő-memória felszabadítása	GRAPHIC CLR

10.2.3. Grafikus/rajzoló utasítások

PIXEL-KURZOR átállítása:
-adott koordinátájú Pontra, vagy
-elmozdítása adott mértékben (+/- előjel kötelező!)
-elmozdítása adott mértékben(+/-) és irány(szög)ben

```
LOCATE <loc>
  ahol:
    <loc>: helyme9határozás
  azaz:
       x,y  : koordináták
       dx,dy: eltolás (+/- Pont)
       hossz,irány: (Pont,fok)
```

A PIXEL-KURZOR Pillanatnyi helyzetének, vagy színének lekérdezése (az argumentum értékétől függően) (függv)

```
RDOT(n)                                    %
  ahol:
    n:  0 - x-koordináta
        1 - y-koordináta
        2 - színre9iszter
```

Téglalap rajzolása <loc> koordinátájú bal felső sarokkal és <loc2> (ennek hiánya esetén a PC) helyzetű jobb alsó sarokkal, adott mértékű jobbra való elforgatással, kitöltve, vagy anélkül

```
BOX [sreg],<loc>[,<loc2>][,szög]
    [,kitöltés]
  ahol:
    sreg: 0-3
    szög: elforgatás jobbra (fok)
    kitöltés: 0 - nincs
              1 - kitöltve
```

<loc> helyzető pont ON/OFF

```
DRAW [sreg],<loc>
```

<loc> és <loc2> végpontokkal definiált egyenes

```
DRAW [sreg][,<loc>] TO <loc2>
```

Poligon rajzolása

```
DRAW [sreg][,<loc>][TO <loc2>...]
```

<loc> középpontú, xh vízszintes, yv függőleges féltengelyű ellipszis szerkesztése adott szögelfordulással

```
CIRCLE [sreg],[<loc>],xh[,yv,,,szög]
  ha yv nincs megadva, kör
  (HI-RES módban, képernyőn)
```

<loc> középpontú ív húzása, kezdő-(ksz) és végszög (vsz) között, egyenes szakaszokkal közelítve az ívet (Poligon)

```
CIRCLE [sreg],[<loc>],xh,[yv],[ksz],
       [vsz],[szög][,növekmény]
  ahol:
    xh,yv: féltengelyek    hossza
    ksz  : kezdőszög (alapért: 0)
    vsz  : végszög (alapért: 360)
    növekmény : szögeltérés két
                ívszakasz között
```

Zárt görbe adott színnel való kitöltése (<loc> a bezárt terület egy pontja)

```
PAINT [sreg][,[<loc>][,func]]
  ahol:
    func: 0 - a megadott 'színű'
              határvonalig színez
          1 - csak 'háttérszínű'
              területet színez
```

10.2.4. Szövegkezelés grafikus módban

Szöveg megjelenítése grafikus üzemmódban, adott karakterpozíciójú (sor,oszlop)mezőben

```
CHAR  sreg,sor,oszlop,szöveg$
    [,inverz]
  ahol:
    sreg :   0-3
    inverz : 0 - normál
             1  - inverz
```

10.2.5. SPRITE-ok

Lehetőség van a grafikus képernyő adott, bal felső és jobb alsó csúcsával jellemzett, téglalap alakú részének tárolására; később vagy más helyen történő megjelenítésére (esetleg feldolgozására).

Képernyő téglalap alakú részének kitárolása vált$-ba

```
SSHAPE vált$,<loc1>[,<loc2>]
```

SSHAPE utasítással tárolt kép adott helyén való megjelenítése

```
GSHAPE vált$,[<loc>][,transz]
  ahol:
    transz: transzformáció
     0 - másolás
     1 - invertálás
     2 - OR-olás képernyővel
     3 - AND-elés
     4 - EOR-olás
```

11. LEMEZ I/O (VC1541 egység):

A 1541-es (floppy-diszk) lemezegység intelligens vezérlővel rendelkezik, így a felé kiadott utasítások végrehajtása az alaprutinban folyó műveletekkel párhuzamosan folyhat, a készülék bonyolult tevékenységek önálló végrehajtására képes. Bizonyos (alább felsorolt) parancsok végrehajtásához elegendő a 15-ös parancs-csatorna megnyitása és használata.

A 15-ös csatorna szolgál az esetleges hibastátusz kiolvasására is. A hibastátusz és hibaüzenet kiolvasása automatikus, az ST, DS, DS$ rendszerváltozók tartalmazzák a szükséges információt.

A lemezes file-ok kezeléséhez általában két csatorna használatára van szükség: a 15-ös parancs-csatornára, és adatátvitel esetén egy vagy több adat-csatornára. Természetesen, az adatátvitel megkezdése előtt mindegyiket külön-külön meg is kell nyitni.

Az I/O műveletek befejeztével a file-okat le kell zárni, különben adatok veszhetnek el. A parancs-csatornát mindig utóljára zárjuk le. Megjegyzendő, hogy a parancs-csatorna lezárása az adat-csatornák lezárásával együtt jár.

11.1. Lemez-Parancsok:

A hajlékony mágneslemez, mint adathordozó karbantartását, kezelését teszik lehetővé (formázás, katalógus-listázás, file-ok létrehozatala és törlése stb.)

11.1.1. BASIC 3.5 lemez-Parancsai:

A BASIC 3.5 új közvetlen lemez-parancsokat vezetett be. Használatukkal jelentősen egyszerűbbé válik a lemezegységek kezelése. Elmarad a 15-ös csatornára való explicit hivatkozás, automatikus a hibacsatorna kiolvasása.

lemez kezdeti formázása (BAM,katalógus és blokkok kialakítása) Kérdés: ARE YOU SURE? Válasz: Y - tovább N - állj	HEADER lemeznév$,Dmegh#[,Iid] [,Uegység#] ahol: id: 2 karakteres azonosító megh#: dual-meghajtónál a belső sorszám (0-1) egység#: 8 / 9 / 10 / 11
lemezkatalógus törlése Kérdés: ARE YOU SURE? Válasz: Y - tovább N - állj	HEADER lemeznév$,Dmegh#[,Uegység#]
lemez-BAM kompressziója (üres helyek felszabadítása)	COLLECT [Dmegh#][,Uegység#]
egész lemezkatalógus vagy file(csoport)ok bejegyzésének listázása memória törlése nélkül	DIRECTORY [Dmegh#][,Uegység#] [,filenév$]
lemez duplikálása duál-egységen (a cél-lemez formázásával!)	BACKUP Dmegh# TO Dmegh#[,Uegység#]
file átmásolása illetve összemásolása (max. 4 db)	COPY [Dmegh#,]régifile$ TO [Dmegh#,] újfile$[,Uegység#]
file átnevezése	RENAME réginév$ TO újnév$ [,Dmegh#] [,Uegység#]
file törlése Kérdés: ARE YOU SURE? Válasz: Y - törlés N - állj	SCRATCH filenév$[,Dmegh#][,Uegys#]
Programfile kimentése memóriából	DSAVE filenév$[,Dmegh#][,Uegység#]
Programfile betöltése memóriába	DLOAD filenév$[,Dmegh#][,Uegység#]
Program-másolat összevetése memóriatartalommal	VERIFY filenév$[,egység#[,típ]] ahol: típ: 8 - BASIC 1 - gépikód

11.1.2. BASIC 2.0 lemez-parancsszövegei

A BASIC 2.0 megérti és végrehajtja a BASIC 2.0 szintaxisa szerint kiadott lemez-parancsokat is. Ezek a 15-ös csatornán küldhetők át, az OPEN utasítás szöveg (string) paramétereként, vagy PRINT# utasítással, stringkifejezés formájában. Mindegyik parancs rövidíthető a kezdőbetűjével is, pl. NEW = N, COPY = C stb.
Az alkalmazható parancsszövegek

lemez inicializálása (BAM, katalógus és blokkok kialakítása)	NEW[meghajtó#]:diszknév,id ahol: id: két-karakteres azonosító meghajtó#: dual-meghajtónál a belső sorszám (0-1)
lemezkatalógus törlése	NEW[meghjt#]:diszknév
lemez-BAM kompressziója	VALIDATE
file átmásolása illetve összemásolása (max. 4 db)	COPY[meghajtó#]:újfilenév= [[meghajtó#]:régifile,...]
file átnevezése	RENAME[meghajtó#]:újnév=réginév
file törlése	SCRATCH[meghajtó#]:filenév
lemezegység alaphelyzetbe hozatala	INITIALIZE[meghajtó#]

11.2. Szekvenciális file-ok kezelése

A szekvenciális file-ok az alább ismertetendő utasítások segítségével logikai szinten úgy kezelhetők, mintha byte-ok folytonos sorozatai lennének, fogyasztásuk attól a blokk-szerkezettől amelyet a D. Fürst-ben ismertetünk. Ezen belül nagyobb logikai egységek (rekordok és más strukturák) kialakítása a felhasználó feladata. A rekord végét CHR$(13) jelzi.

Parancs-csatorna (Pfile#) megnyitása (csak 1.0 végrehajtásának ellenőrzésére - sd. bevezető vagy parancsok kiküldésére).	OPEN Pfile#,egység#,15 ahol egység#: 8 (alaphelyzetben) vagy 9/10/11 (beállítható programmal v. áramkörileg).
Adat - csatorna (afile#) megnyitása	OPEN afile#,egység#,csatorna#, "file-név[,típus][,mód]" ahol típus: PRG - program-file SEQ - szekvenciális REL - relatív-file USR - user-mód (rövidítve P,S,R,U) mód: R - read (olvasás) W - write (írás) A - append (hozzátoldás) M - le nem zárt file olv.
Adat kiírása	PRINT#afile#:kif[<tab>kif...] ahol <tab>: mint PRINT utasításnál. A szóköztől eltérő szeparátorról (CR,,vesszőt) külön gondoskodni kell.
Adat beolvasása (karakter-változónként 1 karaktert olvas be, delimitert nem igényel) - Pfile-ról is.	GET#file#,változó$-lista
Adat beolvasás (típusellenőrzéssel, változónként delimitertől delimiterig) - Pfile-ról is.	INPUT#file#,változó-lista

Megj. az olvasás legalább az utolsó elemet követő CHR$(13)-ig történik. Az így feleslegesen beolvasott adatok elvesznek

File lezarasa CLOSE file#

Meg: Mind az adat-, mind a Parancs-csatornat le kell zarni. A Parancs-csatorna lezarasa az osszes, az adott egysegbe megnyitott adat-file lezarasat is eredmenyezi anelkul, hogy az utolso Puffert kiuritene. Ez outPut-file-oknal adat-elvesztheez vezethet, ezert mindig az adat-csatornakat kell elobb lezarni!

II. Randomkozvetlen elerésü) file-ok kezelése

A random file-ok hasznalata két szintü fizikai kezelést igényel

1. a PARANCS-csatornan keresztül: a lemezterület blokkonkenti kezelése, azaz
   - szabad blokk lefoglalása (allokálása)
   - blokk kiirása Pufferbol
   - blokk beolvasása a Pufferbe
   - felesleges (törölt) blokk felszabaditása
   - blokkok nyilvántartása
   - hibastátusz lekérdezése

2. az ADAT-csatornákon keresztül a Puffer (PufferPointer) kezelése, azaz
   - beirás a Pufferbe
   - kiolvasás a Pufferbol

Parancs - csatorna (Pfile#) megnyitása	OPEN Pfile#,egység#,15[,Parancs$]
Adatcsatorna (afile#) megnyitása (ha a Puffert is kijelöljuk, tartalmát a CLOSE és lemezcsere után is felhasználhatjuk)	OPEN afile#,egység#,acsat#, file-név$+"[Puffer#]" ahol acsat#: adat-csatornaszám file-név$: kötelezően "#" Puffer#: 0 - 4
Blokk beolvasása Pufferbe	PRINT#Pfile#,"BLOCK-READ:"acsat#; meghajtó#;sáv#;szektor# PRINT#Pfile#,"B-R:"acsatorna#; meghajtó#;sáv#;szektor#
Pufferből való kiolvasás	INPUT#afile#,változó-lista GET#afile#,változó$-lista

A Puffer kurrens (utóljára beirt, ill. következő kiolvasásra kerülő) byte-jára mutató Pointert ilyenkor a rendszer automatikusan állitja. Ugyanakkor a Pointer tetszés szerint is beállitható, ezzel a blokkon belül is véletlenszerü elérést valósithatunk meg:

Puffer-Pointer állitása, blokkon belüli adott byte-ra	PRINT#Pfile#,"BUFFER-POINTER:"acsat#; byte-Pozició PRINT#Pfile#,"B-P:"acsatorna#;byte-Poz

Egy blokkot első felirása előtt le kell foglalni a BAM-ban ahhoz, hogy biztonságosan használható legyen a lemez.
FIGYELEM: A VALIDATE Parancs kiadása azonban random-file-t tartalmazó lemezre akkor is **tilos!**

Blokk lefoglalása	PRINT#Pfile#,"BLOCK-ALLOCATE:"meghajtó#; sáv#;szektor# PRINT#Pfile#,"B-A:"meghjt#;sáv#;szektor#

Ha a blokk már foglalt, a DOS 65-ös hibakódot ad vissza a hibacsatornán, és egyúttal az első szabad blokk címét, ami aztán felhasználható irásra, pl

```
1000 PRINT#15,"B-A:"0;sáv;szektor
1010 INPUT#15,A,B$,C,D
1020 IF A=65 THEN sáv=C  szektor=D: GOTO1000
1030 PRINT#15,"B-W:"15.0.sáv;szektor
```

Puffer feltöltése	PRINT#afile#,kif[<tab>kif...]
Puffer kiírása (feltöltés után)	PRINT#pfile#,"BLOCK-WRITE "acsatorna#,megnajtó#,sáv#,szektor# PRINT#pfile#,"B-W "acsatorna#,meghajtó#,sáv#,szektor#

11.4. Fizikai-blokk kezelő utasítások

Az ismertetett utasítások esetében a visszaolvasott blokk hossza egyezik a kiírt blokkéval. B-W-vel való kiíráskor ugyanis a Puffer-Pointer tartalma (blokk kurrens mérete) a 0-byteba kerül. B-R-rel való olvasáskor az ebben jelzett hossz elérésekor FILE VÉGE státuszt kapunk az ST rendszerváltozóban. Lehetőség van azonban a logikai blokkmérettől függetlenül a teljes fizikai blokk (256 byte) kiírására ill. visszaolvasására.

Fizikai blokk beolvasása	PRINT#pfile#,"U1 "acsatorna#,meghajtó#,sáv#,szektor# PRINT#pfile#,"UA "acsatorna#,meghajtó#,sáv#,szektor#
Fizikai blokk kiírása	PRINT#pfile#,"U2 "acsatorna#,meghajtó#,sáv#,szektor# PRINT#pfile#,"UB:"acsatorna#,meghajtó#;sáv#;szektor#

Ez az utasítás nem írja felül a Puffer-Pointer lemezen tárolt előző értékét!

11.5. Relatív file-ok kezelése

Ennél a fileszervezési típusnál a rekordok (max. 720 db) a fileon belüli sorszámukkal címezhetők. Ez azzal jár, hogy a fizikai rekordok fix hosszúságúak, a logikai rekord hossza célszerűen ezzel egyenlő vagy kisebb. Mivel az I/O utasítások a rekordhatár átlépésekor hibajelzést nem adnak, a programozó - ha megfelelően kezeli - használhat hosszabb rekordokat is. A kiíró utasítás a rekordba való íráskor a rekord végéig torol. A DOS egy-egy rekord kiírásakor az összes annál kisebb sorszámú rekordot is létrehozza. Célszerű tehát - a használatot gyorsítja - ha a várható legmagasabb sorszámú rekordot a file létrehozásakor felírjuk.

A relatív file használata 3 fő fázisból áll:
- a file létrehozása, rekordméret kijelölése
- a file kialakítása (rekord-tartomány létrehozatala)
- file írása,olvasása,bővítése,módosítása

Parancsfile megnyitása	OPEN pfile#,egység#,pcsatorna#
adatfile létrehozása, rekordmeret megadása	OPEN afile#,egység#,acsatorna#,filenév$+",L,"+CHR$(rekordhossz)
adatfile további megnyitásai	OPEN afile#,egység#,acsatorna#,filenév$
Pozicionálás a rekordra és azon belül a kívánt byte-ra (minden I/O előtt kötelező)	PRINT#pfile#,"P"+CHR$(acsatorna#)+CHR$(rek#-Lo)+CHR$(rek#-Hi)+CHR$(byte-Poz#) ahol: rek#-Lo/rek#-Hi: rekord-sorszám alsó / felső byte-ja byte-Poz#: kezdő byte-Pozíció sorszáma
file írása / olvasása: u.a. mint szekvenciális file-oknál	PRINT#afile#,kif[<tab>kif....] INPUT#file#,változó-lista GET#file#,változó$-lista
file lezárása	CLOSE file#

11.6. Floppy-memória kezelése

[illegible] a floppy-vezérlő memóriájából olvasni, ill. [illegible] és azt végrehajtatni. Az ezt [illegible] I/O utasítások

[illegible] betoltése a [illegible] a Pufferbe és végrehajtása a 0-ás relatív címtől kezdve	PRINT#Pfile#,"BLOCK-EXECUTE:"acsat#;me9hajtó#;sáv#;szektor# PRINT#Pfile#,"B-E:"acsat#;me9hajtó#,sáv#;szektor#
1 byte olvasása a floppy memóriájából (ROM vagy RAM)	PRINT#Pfile#,"M-R:"+CHR$(bytecím-Lo) + CHR$(bytecím-Hi)
max. 34 byte beírása a floppy memóriájába	PRINT#Pfile#,"M-W:"+CHR$(bytecím-Lo) + CHR$(bytecím-Hi),karakterszám,adat$
Program végrehajtatása floppy memóriájában	PRINT#Pfile#,"M-E:"+CHR$(kezdőcím-Lo)+ CHR$(kezdőcím-Hi)

További USER utasítások (a lemezegység memóriájában levő címekre való ugrást teszik lehetővé)

jump $0500	PRINT#Pfile#,"U3"	(vagy "UC")
jump $0503	PRINT#Pfile#,"U4"	(vagy "UD")
jump $0506	PRINT#Pfile#,"U5"	(vagy "UE")
jump $0509	PRINT#Pfile#,"U6"	(vagy "UF")
jump $050[illegible]	PRINT#Pfile#,"U7"	(vagy "UG")
jump $050[illegible]	PRINT#Pfile#,"U8"	(vagy "UH")
jump $fffa (NMI vektor)	PRINT#Pfile#,"U9"	(vagy "UI")
[illegible]	PRINT#Pfile#,"U;"	(vagy "UJ")
[illegible]	PRINT#Pfile#,"UI+"	
[illegible]	PRINT#Pfile#,"UI-"	

12. NYOMTATÓ (VC1525 / MPS801 / MPS803 mátri printer) KEZELÉSE

A COMMODORE mátri nyomtatók intelligens perifériák, saját belső puffereszközzel rendelkeznek. Több különböző üzemmódjuk van, melynek beállítása az OPEN utasítással, továbbá a PRINT# utasításban kiadott vezérlő-karakterekkel / karakterszekvenciákkal történhet.

A nyomtató puffere 90 byte-os, ebből legalább 1 byte-ot elfoglal az aktuális vezérlő-karakter szekvencia. Az effektív nyomtatás akkor kezdődik, ha
- [illegible] nyomtatása kezdődik
- a puffer betelik
- 480 pontnyi hely felhasználásra került a sorban
  (a sor tele van)

Ha az információ nem fér a sorba, a következő sorban folytatódik nyomtatása.

A nyomtatási sorosség karakteres módban 6 sor/inch, grafikus módban 9 sor/inch.

A 3.fejezetben általánosságban már ismertetett periféria-vezérlő parancsok és I/O utasítások speciális formái:

12.1. Kiírás nyomtatón - I/O utasítások:

Printer-file megnyitása	OPEN file#,egység#,üzemmód ahol: egység#: 4 vagy 5 üzemmód: (alapállapot a file lezárásáig): 0 - nagybetű/grafikus 7 - nagybetű/kisbetű
Kiírás nyomtatóra	PRINT#file#,[[<cntrl>]kif...] ahol: <cntrl> , - standard tabulálás ; - szóköz-elnyomás SPC(kif#) - kif# szóköz beszúrása TAB(kif#) - kif#-ik pozícióra tab <vezérlő-szekvencia> - lsd. 12.2.
Puffer kiürítése file lezárása előtt	PRINT#file#
Printer-file zárása	CLOSE file#
Képernyőre menő információ nyomtatóra irányítása (hallgató üzemmód beállítása) és kif. nyomtatása:	CMD file#[,kif[<cntrl>kif...]]
Printer csatorna lezárása hallgató üzemmódban (hallgató üzemmód felfüggesztése)	PRINT#file#

Megj: a printerfile-t a CMD kiadása előtt meg kell nyitni, a CMD után következhet(nek) a LIST vagy PRINT utasítás(ok). A file lezárása előtt egy üres PRINT# parrancsot kell kiadni a csatorna lezárására. Ez után következhet a CLOSE utasítás.

12.2. Nyomtató-vezérlő karakterek:

A <vezérlő szekvencia>: egy vagy több karakterből álló sorozat, amely az utána következő string értelmezését, vagy az adott PRINT# utasítás végrehajtását, esetleg a file lezárásáig terjedő nyomtatást módosíthatja.

A lehetséges kombinációk

Hatás	Szekvencia	Hatókör
azonnali soremelés	CHR$(10)	egyszeri
azonnali kocsi-vissza	CHR$(13)	egyszeri
dupla vonalvastagság	CHR$(14)	CHR$(15) kiadásig
normál - karaktermód	CHR$(15)	CHR$(14) kiadásig
kisbetű/nagybetű mód	CHR$(17)	PRINT# végéig vagy CHR$(145) kiadásig
nagybetű/grafikus mód	CHR$(145)	PRINT# végéig vagy CHR$(17) kiadásig
inverz-karakter mód	CHR$(18)	CHR$(146) kiadásig
inverz-módból vissza	CHR$(146)	CHR$(18) kiadásig
az nn kétkarakteres numerikus string által definiált karakterpozícióba tabulál	CHR$(16)+"nn"	egyszeri
bitképes-módba kapcsolás	CHR$(8)	PRINT# végéig vagy CHR$(15)-ig
a bitkép# által reprezentált pont-oszlop ismétlése rep#-szer (csak bitképes módban)	CHR$(26)+ CHR$(rep#)+ CHR$(bitkép#)	egyszeri
a 4 karakteres vezérlő sorozatot követő stringet a bpoz által definiált pont-pozícióba (1-480) tabulálja (csak bitképes módban)	CHR$(27)+ CHR$(16)+ CHR$(bpoz-low)+ CHR$(bpoz-high)	egyszeri
saját tervezésű 6*7 pontból álló karakter nyomtatása. Karakterpozíciónként 6, a karakterkép oszlopaiból képzett szám 128-al megnövelt értékéből CHR$-ral konvertált karaktert kell a vezérlőkarakter után elküldeni.	CHR$(8)+ CHR$(1.oszlop)+ CHR$(2.oszlop)+	CHR$(15) kiadásig

13. RS232 INTERFACE HASZNÁLATA:

Az RS232 interface széleskörűen használható különböző nem-COMMODORE gyártmányú perifériák (RS232S vagy RS232C interface-szel csatlakozó nyomtatók, stb) továbbá távadatátviteli feladatok megoldására megfelelő csatoló egység(modem) felhasználásával.

A' C16-nál az RS232 kezelése is a TED-be van integrálva, ezért bár nincs a C64-hez hasonló user-Portja, az RS232 típusú kommunikáció a soros-buszon keresztül, vagy közvetlenül az expansion-Portra csatlakozó speciális illesztőegységgel valószínűleg lehetséges.

Az RS232 csatorna Pufferezéséhez a BASIC-interPreter egy darab 64 byte-os Puffert használ. Ezért egyszerre legfeljebb egy RS232 csatornát szabad megnyitni.

13.1. I/O utasítások:

RS232 csatorna megnyitása

```
OPEN file#,2,0,Parancs$
```

ahol:

```
Parancs$:
   kontroll.regiszter-byte+
   Parancs.regiszter-byte

kontroll.regiszter-byte:
   bit(3-0)  átviteli sebesség
                0 - 3-ik és 4-ik byte-
                    ban van definiálva
                1 - 50     baud
                2 - 75     baud
                3 - 110    baud
                4 - 134.5  baud
                5 - 150    baud
                6 - 300    baud
                7 - 600    baud
                8 - 1200   baud
                9 - 1800   baud
               10 - 2400   baud
               11 - 3600   baud
               12 - 4800   baud
               13 - 7200   baud
               14 - 9600   baud
               15 - 19200  baud
   bit  (4):  0 - külső óra
              1 - adatátvitel
   bit(6-5): szóhossz
              0 - 8  bit
              1 - 7  bit
              2 - 6  bit
              3 - 5  bit
   bit (7)  stoP-bitek száma
              0 - 1  stoP-bit
              1 - 2  stoP-bit

Parancs.regiszter-byte:
   bit(1-0): receiver-vezérlés
     bit 0 :   receiver be(1)/ki(0)
               DTR     alacs./magas
     bit 1 :   IRQ       be(0)/ki(0)
               (ST reg, bit#3)
   bit(3-2): transmitter-vezérlés
                IRQ   RST transmitter
          00  ki  magas     inakt.
          01  be  alacs.     aktív
          10  ki  alacs.     aktív
          11  ki  alacs.     BRK
   bit(4):   0 - normál
             1 - echo
   bit(7-5): Paritás-funkciók
             1 - Páratlan Pár. (i/o)
             3 - Páros Pár. (in/out)
             5 - Jelátvitel
                 Paritásell. nincs
             7 - helyköz átvitele
                 Paritásell. nincs
           Páros - Paritásellenőrzés
                   Paritásbit-generálás
                                 nincs
```

Adatok kiolvasása a pufferből	GET#file#,változó$ INPUT#file#,változó-lista használata nem ajánlott a vonali Protokollokban szereplő speciális karakterek miatt (lsd. [2] 6.16.)
Írás pufferbe	PRINT#file#,kifejezés$-lista vagy CMDfile#
File lezárása	CLOSE file#

13.2. Az RS232 állapotregiszterben megjelenő státuszok
(kiolvasáskor azonnal törlődnek)

bit(0) - Paritáshiba
bit(1) - szerkesztési hiba
bit(2) - input puffer túlcsordulás
bit(3) - input puffer üres
bit(4) - CTS-jel nem jött
bit(5) - nem használt
bit(6) - DSR-jel nem jött
bit(7) - vonalszakadás

13.3. RS232 memória-címek

Vezérlő információk	név	dec	hex
I/O regiszter		64768	$fd00
státusz-regiszter		64769	$fd01
kontrol-regiszter		64770	$fd02
Parancs-regiszter		64771	$fd03
Pszeudo státusz-regiszter	RSSTAT	2004	$07d4
BASIC ST(státusz) változó	ST	144	$0090
Input:			
input puffer (64 byte)	RIBUF	1015	$03f7
		-1078	-$0436
input FIFO-puffer elejének byte-indexe	RIDBS	2002	$07d2
input FIFO-puffer byte-számláló	RIDBE	2003	$07d3
Output:			
output puffer (1 byte)	ROBUF	1997	$07cd
output byte üres flag		1998	$07ce
Munkaterületek és belső rutinok:			
RS232 regiszterek		64768	$fd00
(TED vezérlő regiszterek)		-64782	-$fd10
RS232 munkaterületek		1997	$07cd
		-2008	$07d8
		252	$00fc
		-253	$00fd
interrupt rutin			$ea5b
1 byte olvasása (A regiszterbe)			$eaf1
RS232 CHKIN, CKOUT			$eb25
1 byte kiírása (A regből $07cd-be)			$eb37
RS232 OPEN, CLOSE	]		$eb46

14. KAZETTÁS SZALAGEGYSÉG (DATASETTE 1531) KEZELÉSE

A DATASETTE kazettás szalagegység programok tárolásán túlmenően alkalmas adatok tárolására is. Az adatok elérése azonban - a mágneslemezzel (diszk) ellentétben csak szigorúan szekvenciálisan (sorosan) történhet.

Tekintettel arra, hogy a szalagegységen folyó I/O alatt a megszakítások le vannak tiltva, az adatátvitel ideje alatt a képernyő nem használható (üres).

Az I/O a szalagegység esetében is pufferezett, a puffer hossza 192 byte.

A standard tárolási mód esetében a byte-ok tárolása a következő módon történik:
- a byte-okat egy bytejelző jelsorozat vezeti be, azaz:
  egy 846/us majd egy 420 /us négyszögjel(LH)
- a bytejelzőt követi a 9 (8 adat, 1 paritás) bit:
  1 420/us majd 208/us négyszögjel
  0 208/us majd 420/us négyszögjel

A kódolás páros paritású.

14.1. A szalagos file-ok szerkezete

PROGRAM-file:

gap 8s (törlés)	fejléc 4s	gap 3.4s (törlés)	Program

FEJLÉC-blokk(program-file)

1	kezdőcím-Lo	kezdőcím-Hi	végcím-Lo	végcím-Hi	Prognév

FEJLÉC-blokk(gépi-kódú program-file):

3	kezdőcím-Lo	kezdőcím-Hi	végcím-Lo	végcím-Hi	Prognév

ADAT-file

gap	fejléc	gap	blokk#1 191 byte	gap	blokk#2 191 byte	gap	stb

FEJLÉC-blokk(adat-file):

4	kezdőcím-Lo 60	kezdőcím-Hi 3	végcím-Lo 252	végcím-Hi 3	filenév

ADAT-blokk

2	adatok	adat vége 0

ADAT-VÉGE-címke

5	kezdőcím-Lo	Kezdőcím-Hi	végcím-Lo	végcím-Hi	filenév

14.2. I/O utasítások

File megnyitása : (hatására a fejléc a kazetta-Pufferbe kerül)
```
OPEN file#,1,csatorna#,[file-név$]
  ahol:
    csatorna#: 0 - olvasás      (inPut)
               1 - írás         (outPut)
               2 - írás (outPut), file végén End-Of-File címke
```

Kiírás szalagra :
```
PRINT#file#,[kif[<tab>]...]
  ahol:
    <tab>: ; - szóköz-elnyomás
           , - tabulálás (10 oszloponként)
           SPC(kif#) - kif# db szóköz beszúrása
           TAB(kif#) - Pozicionálás a kif#-ik oszlopra
```

Megj: A mágnesszalagra egyes karakterek nem kerülnek kiírásra (CHR$(10) és CHR$(29)) , a CHR$(0)-nak pedig speciális jelentése van (End-Of-Data, file-vége)

Beolvasás szalagról
```
GET#file#,változó$-lista
INPUT#file#,változó-lista
```

Megj:Az INPUT# utasítás használatakor gondot okozhatnak az idézőjel és kettőspont karakterek.

Szalagfile lezárása
```
CLOSE file#
```

14.3. Speciális I/O-k (Program tárolása/betöltése)

Program tárolása szalagra
```
SAVE Programnév$,1[,kód#]
  ahol
    kód#: 0 - kiírás BASIC Programként
          1 - kiírás gépi-kódú Program-ként (3-as fejléc-blokkal)
          2 - kiírás BASIC Programként, végén SZALAG-VEGE jellel
          3 - kiírás gépi-kódú Program-ként, SZALAG-VEGE jellel
```

Program betöltése szalagról
```
LOAD Programnév$,1[,kód#]
  ahol
    kód#: 0 - betöltés BASIC-Programként (3-as fejlécű file gépi kódú Programként)
          1 - gépi kódú Programként (a fejlécben megadott címre)
```

14.4. Fontosabb memóriacímek:

tartalom	név	dec	hex
szalag-Puffer Pointer	FBUFPT	182,183	$b6,$b7
szalagPuffer		819-1010	$333-$3f2
blokk ellenőrző-összeg .		245	$f5
első/második kiírás flag		247	$f7
fejléc(címke)-típus		248	$f8
olvasás hibatáblázat		1079-1138	$437-$472
olvasás hibaszámláló		1974	$7b6
szalag kiírás számláló		1011,1012	$3f3,$3f4
szalag beolvasás számláló		1013,1014	$3f5,$3f6
Puffer/tár kapcsoló		1968	$7b0
munkaterület		1969-1996	$7b1-$7cc
időzítési konstansok		1976-1978	$7b8-$7ba
Program kezdete	SAL	178,179	$b2,$b3
Program vége	EAL	180,181	$b4,$b5

15. BOTKORMÁNY

A COMMODORE-16 két speciális, 8 pólusú játék-csatlakozóval rendelkezik, amelyekhez botkormány (joystick) csatlakoztatható. Állapotuk kiolvasása BASICból JOY függvénnyel vagy gépi kódú rutinokkal történhet. Analóg bemenet (Potenciométer, Paddle) kezelésére nincs lehetőség.

15.1. A botkormány használata

A botkormány 5 kapcsolóval rendelkezik: 4 irány (0-as:Észak/fel, 1-es:Dél/le, 2-es Nyugat/balra, 3-as:Kelet/jobbra) és a TŰZ (4-es kapcsoló). Ezekből -a botkormány természetéből következően- egyszerre maximum 2 szomszédos irány és a TŰZ-kapcsoló nyomható be. A két CIA vagy 2-es és 8 vagy 1-es) botkormány kapcsolói a billentyűmátrix-szal közösen a TED-chip $ff08 regiszterére vannak kötve, bit#6 ...#1 ill bit#7(Joy#2): TŰZ, bit# 3-0: megfelelő irány-kapcsolók.
A kapcsok 7-0 bitjei alaphelyzetben 1-esek, a lenyomott kapcsoló a megfelelő bitet 0-ra állítja.

A kapcsolók lekérdezése a következőképpen történhet:

BASIC-ből

A JOY függvény szintaxisa:

JOY(kif#), ahol kif#: joystick-szám (1 v. 2),

a függvényérték pedig:

0 - semmi
1 - É 2 - ÉK 3 - K 4 - DK
5 - D 6 - DNY 7 - NY 8 - ÉNY

a TŰZ-gomb lenyomása esetén a fenti értékekhez 128 adódik hozzá!

Gépi-kódú rutinból:

Az interrupt letiltása (SEI utasítás) után a botkormány#1-hez a TED $ff08 (65288) regiszterének 2-es bitjén, a botkormány#2-höz az 1-es biten kívül minden bitet ON-ba állítunk ($fb ill. $fd beírásával), majd kiolvassuk a regisztert. A kiolvasott érték a kiválasztott botkormányra fog vonatkozni.

Miután a billentyűzet-mátrix ugyanezen regiszterekre van kötve, ez azt is jelenti, hogy a botkormányok a billentyűzetről szimulálhatók.

Az alábbiakban a lehetséges megfeleltetéseket ismertetjük.

A JOY függvény számára mindkét botkormány minden további nélkül helyettesíthető a következő billentyűkkel:

Észak fel	ÉK	Kelet Jobbra	DK	Dél le	DNY	Nyugat balra	ÉNY	TŰZ
1	1+2	2	home+2	home	home+ctrl	ctrl	ctrl+1	spc

további lehetőségek

1	1+2	2	home+2	home	home+ctrl	ctrl	ctrl+1	Q

15.2. Felhasznált memóriarekeszek (TED-regiszterek):

Tartalom	cím dec	cím hex
TED $08 regiszter bit 1: botkormány#2 kiválasztás (OFF) bit 4: botkormány#2 tüzelőgombja bit 3-0: botkormány#2 irány-kapcsolói	65288	$ff08
TED $08 regiszter bit 2: botkormány#1 kiválasztás (OFF) bit 4: botkormány#1 tüzelőgombja bit 3-0: botkormány#1 irány-kapcsolói	65288	$ff08

16. HANG-KIMENET KEZELESE

16.1. Általános információk

A TED 7360 végzi a hanggeneráláshoz szükséges műveleteket is. Két hanggenerátort tartalmaz (sorszámuk 1-2), amelyek általában egymástól függetlenül vezérelhetők, de egymással szinkronizálhatók ill. csatolhatók is. A hanggenerátorok paraméterezése és vezérlése a TED kontroll-regiszterein keresztül történik.

A TED hang-regisztereinek kiosztása

(Fő báziscím, jelölése továbbiakban "b") b= 65280 $ff00

Reg#	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reg#
+0	timer#1-LO								00
+1	timer#1-HI								01
+2	timer#2-LO								02
+3	timer#2-HI								03
+4	timer#3-LO								04
+5	timer#3-HI								05
	.								
	.								
+14	oszcill#1 frekvencia-LO								0e
+15	oszcill#2 frekvencia-LO								0f
+16							oszc#1-frq.HI		10
+17	sound reload	han9#2 zaj	han9#2 négysz	han9#1 bekap	hangerő				11
+18							oszc#2-frq.HI		12
	.								
	.								
+62	ROM-engedélyezés								3e
+63	ROM-letiltás								3f

16.2. Az előállított hang jellemzői:

HANGMAGASSAG(frekvencia): 2 byte (low/high) (10 bit)

oszcillátor#1 b+14 , b+16 $ff0e,$ff10
oszcillátor#2 b+15 , b+18 $ff0f,$ff12

A TED hangterjedelme kb. 5 oktávot fog át (A#2-D#8, 109Hz - 110KHz frekvencia, 1-1022 oszcillátor-frekvencia tartományban). Oktávonként 12 hang van: C, C#,D,D#,E,F,F#,G,G#,A,A#,H. Az egyenlően temperált hangskálán két szomszédos hang frekvencia-viszonya:

$$f_{i+1} = f_i * 2^{(1/12)}$$

Két szomszédos oktáv hangjainak frekvencia-aránya:

$$f_{j+1} = f_j * 2$$

Ha ettől eltérő frekvenciákat akarunk generálni, a hangmagasság és az oszcillátor-frekvencia (a regiszterben levő bináris szám) közötti összefüggés:

$$f_{hang} = 1024 - (111840.45 / f_{oszc}) \text{ Hz}$$

A két byte ennek megfelelően

$$f_{high} = INT(f_{oszc} / 256) \qquad f_{low} = f_{oszc} - 256 * f_{high}$$

HANGERŐ: egy közös regisztere van v=b+17 65297 $ff11

9 fokozatban állítható (0 - 7), Pl. a következő utasítással (bit 3-0):

```
POKE v,(PEEK(v)AND240)OR hangerő
```

vagy az ezzel egyenértékű (és egyszerűbb)

```
VOL hangerő
```

utasítással.

HULLÁMALAK: csak a 2. oszcillátornál állítható:
v=b+17 65297 $ff11

A választható hullámalakok:

négyszögjel:	zongora,klarinét	(bit#5)
fehér-zaj:	zajok,zörejek	(bit#6)

Megj ha bit#7=1, a hangerő változtatásával befolyásolható egyenáramú jel keletkezik ez felhasználható Pl. beszédszintetizálására.

16.3. A hanggenerátorok (oszcillátor) megszólaltatása
v=b+17 65297 $ff11

oszcillátor#1	bit#4
oszcillátor#2	bit#5

BASIC-ből

```
SOUND szólam,frekvencia,tartam
```

ahol:

szólam: 1, 2 vagy 3 (2.oszcill., fehér zaj)
tartam: időtartam, 1/60 másodpercben
frekvencia: oszcillátor-frekvencia

vagy a megfelelő POKE utasítással:

```
POKE v,(PEEK(v) OR (oszc# * 16))
```

Természetesen ez utóbbi esetben a frekvencia beállításáról és az időtartam méréséről (a C64-hez hasonlóan) külön kell gondoskodni.

16.4. Frekvenciaértékek

okt	hang	oszcillátor frekv			okt	hang	oszcillátor frekv		
		f	low	hi			f	low	hi
	a	7	7	0	6	c	917	149	3
	a#	68	68			c#	923	155	
	h	114	114			d	928	160	
3	c	170	170			d#	934	166	
	c#	219	219			e	939	171	
	d	263	7	1		f	943	175	
	d#	307	51			f#	948	180	
	e	346	90			g	952	184	
	f	384	128			g#	956	188	
	f#	419	163			a	960	192	
	g	453	197			a#	964	196	
	g#	486	230			h	967	199	
	a	515	3	2	7	c	970	202	
	a#	543	31			c#	973	205	
	h	571	59			d	976	208	
4	c	597	85			d#	979	211	
	c#	620	108			e	981	213	
	d	643	131			f	983	215	
	d#	664	152			f#	986	218	
	e	685	173			g	988	220	
	f	703	191			g#	990	222	
	f#	721	209			a	992	224	
	g	738	226			a#	994	226	
	g#	754	242			h	995	227	
normál	a	769	1	3	8	c	997	229	
	a#	783	15			c#	998	230	
	h	797	29			d	1000	232	
5	c	810	42						
	c#	822	54						
	d	833	65						
	d#	844	76						
	e	854	86						
	f	863	95						
	f#	872	104						
	g	881	113						
	g#	889	121						
	a	896	128						
	a#	903	135						
	h	910	142						

1. GRAFIKUS LEHETŐSÉGEK

A C-16 BASIC 3.5-ø a C-64 (2.0-ás) BASIC-jével ellentétben számos grafikus parancsot tartalmaz, amelyek a felhasználó számára szinte felesleges sé teszik az egyébként is lassú PEEK/POKE utasítások bonyolult használatát. Mivel bizonyos lehetőségek (Pld. finom-scroll, képernyőmemória váltás, stb.) csak gépi-kódú rutinokkal, vagy POKE/PEEK utasításokkal használhatók ki, az alább ismertetendő elvek és címek mégis hasznosak lehetnek azok számára, akik saját grafikus elképzeléseiket akarják megvalósítani, esetleg éppen gépi-kódban.

1.1. A TED grafikus regiszterének kiosztása báziscím ba=65280 $ff00

offs	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reg#
+0	timer#1-LO								00
+1	timer#1-HI								01
+2	timer#2-LO								02
+3	timer#2-HI								03
+4	timer#3-LO								04
+5	timer#3-HI								05
+6	teszt	extnded color	HI-RES mód	screen off	24/25 sor	függőleges scroll eltolás			06
+7	revers mód	PAL/ NTSC	TED OFF	MULTI mód	38/40 oszlop	vízszintes scroll eltolás			07
+8	billentyűzet matrix oszlopainak flag-jei								08
+9	IRQ	timer3	-	timer2	timer1	fényc.	raszter	-	09
+10	IRQ	timer3	-	timer2	timer1	fényc.	raszter	raszter.bit#8	0a
+11	raszter-regiszter								0b
+12						kurzor-poz.HI			0c
+13	kurzor-pozíció								0d
+18		grafikus képernyő helye			JEL-gen ROM/RAM				12
+19	karakter-generátor címe							ROM-ON	13
+20	video-mátrix címe								14
+21		fényerő			háttérszín#0 (képernyő)				15
+22		fényerő			háttérszín#1				16
+23		fényerő			háttérszín#2				17
+24		fényerő			háttérszín#3				18
+25		fényerő			keret színe				19
+26						reg#1b 9-8bit			1a
+27	aktuális sor kezdő karakterpozíció 7-0 bitjei								1b
+28								reg#1d 8.bit	1c
+29	aktuális video-sor sorszáma								1d
+30	vízszintes pozíció 8-1 bitjei								1e
+31		villogás-időzítő			karakteren belüli sor-számláló				1f
+62	ROM-engedélyezés								3e
+63	ROM-letiltás								3f

Megjegyzendő, hogy a TED egyes regiszterei minden képfrissítési (raszter-interrupt) ciklusban felülíródnak (RAM-ból vagy ROM-ból újra töltődnek), ezért BASIC-ből való felülírásuk hatástalan, és gépi-kódból is az interrupt-kezelés módosítását igényelné. Hatásos lehet viszont esetlegesen RAM-beli forrásuk (Pszeudo-regiszter) tartalmának módosítása, akár BASIC-ből is. A következőkben ezért (ahol van ilyen) ezeket a Pszeudoregisztereket is megadjuk.
A fentiekből is következik, hogy az alábbiakban szereplő POKE utasítások inkább az egyértelmű definíciót, vagy a követendő algoritmus meghatározását szolgálják, mintsem kódolási mintákat, hiszen arra gyakran közvetlen BASIC-utasítások is vannak (lsd. BASIC fejezet), bár többnyire a megadott formában is használhatóak.

A TED-7360 4 területet lát egyszerre:

a. 2Kb terület (BLOKK) a VIDEO-mátrix (SZÍN és KÉP) tárolására

b. A (többnyire) ROM-ban levő JEL-matrixot (2 Kb) 53248 $d000
(karakter-generátor) -55295 $d7ff

c. A 8000 byte nagyságú grafikus-képernyő-mátrixot (BIT-mátrix) grafikus módban

d. A $ff00-n kezdődő vezérlő regisztereket ba=65280 $ff00
-65343 $ff3f

1.2. Választható üzemmódok:

- karakteres:
  - -írógép mód: kis és nagybetűk, írásjelek
  - -normál mód: nagybetűk, írás- és grafikus jelek
- karakteres, többszínű háttérrel
- nagy felbontóképességű (HI-RES) grafika
- nagy felbontóképességű grafika + 5 sor szöveges rész
- többszínű HI-RES grafika (MULTI-mód)
- többszínű (MULTI) mód + 5 sor szöveges rész

Különlegesség a C-64-hez képest, hogy az egyes színeken belül 8 különböző árnyalat állítható elő. Meg kell említsük még az egyszerű osztott (grafikus és karakteres) képernyő-kezelési és ablakkezelési lehetőséget.

2. KARAKTERES ÜZEMMÓD

BASIC-ből való kényelmes kezelésére szolgáló utasításokat lásd a BASIC fejezet 10.1 pontjában.

JELLEMZŐK		cím /offset dec	hex
25 sor * 40 jel 1 jel = 8 * 8Pont (oktett)			

ALAPADATOK

* VIDEO-mátrix: 2 kbyte-os terület, BLOKK-határra igazítva, ezen belul helyezkedik el a
  * SZÍN-mátrix: a VIDEO-mátrix első 1000 ($3e8) byteja, a színkód (0-15) a jobb félbyte-ban van. A 6-4 bitek a fényerő értékét tartalmazzák, míg a 7. bit a karakter villogtatását kapcsolja (1: ON).
  * KÉP-mátrix: 1000 ($3e8) byte-os folytonos terület, 1 byte/jel, Kbyte (modulo 1024) határra igazítva

  Mindkét mátrix sorfolytonosan képezi le a képernyőt.

* JEL-mátrix: 2Kb-os BLOKK

* háttér-szín regiszter: 0-15 a jobb félbyte-ban 65301 $ff15

* keret-szín regiszter: 0-15 a jobb félbyte-ban 65305 $ff19

* kurrens karakter színkódja: 1339 $053b

* háttérszín a kurzor alatt: 2029 $07ed

* a kurzor aktuális Pozíciója (a karakterPozíciókat 0-999-el sorfolytonosan számozva), kiolvasható, ill. felülírásával a kurzor elmozdítható: (bit 9-8) 65292 $ff0c
(csak gépi kódból kezelhető) (bit 7-0) 65293 $ff0d
BASIC-ből is kezelhető Pszeudo-regiszterek: sor : 205 $cd
oszlop: 202 $ca

2.1. Standard mód:

* KÉP-mátrix: 3072 $0c00
- karakter elhelyezése a képernyőn:
POKE 3072 + 40*sor + oszlop,karakterkód

* SZÍN-mátrix: 2048 $0800
- karakter színének beállítása:
POKE 2048 + 40*sor + oszlop,színkód+
+16*fényerő+128*villogás
(sor: 0-24, oszlop: 0-39)
(karakterkód: normál: 0-127
inverz: 128-255)
(villogás: 0 - normál
1 - villogó)
(fényerő: 0-7 (min-max))
(színkód: 0-15! lsd. A. FÜGGELÉK)

* JEL-mátrix: ROM-ban található:

	ROM
Nagybetűk	53248 $d000
Grafikus jelek	53760 $d200
Kisbetűk	54272 $d400
Nagybetűk+grafikus jelek	54784 $d600

2.2. User mód:

* JEL-mátrix RAM-ban

-JEL-mátrix átkapcsolása RAM-ba: bit 2 -> 0 n=65298 $ff12

```
POKE n,PEEK(n) AND 251
```

-kijelölése a RAM-on belül(BLOKK-határon):
kijelölő regiszter 3-1 bitjein:

```
POKE d,(PEEK(d) AND 240) OR 9
```

d=65299 $ff13

ahol 9=cím/256 (alapértelmezés: 9=208)

* VIDEO-mátrix kijelölése:

-helye karakteres módban nem változtatható, csak az interrupt rutin módosításával

* Hozzáférés a standard JEL-mátrixhoz:
Mivel a PEEK függvény mindig a RAM-ot olvassa, a karaktergenerátor ROM-hoz nem férhetünk hozzá vele. Lehetséges megoldások:

a) ROM átmásolása RAM-ba a MONITOR-ral
b) SYS utasítással meghívni a KERNAL ROM-olvasó rutinját (lsd. GÉPIKÓD, 4.3 Pont), Pld.:

```
10 A=53248 :REM ROM JEL-MATRIX KEZDOCIME
20 FOR I=0 TO 4088 STEP 8: FOR J=0 TO 7
30 K=A+I+J:  GOSUB 500
40 NEXT J:   NEXT I
50 END
500 CH%=INT(K/256): REM CIM HI-BYTE
510 CL%=INT(K-256*CH%): REM LO-BYTE
520 POKE 2036,0:     REM .Y-BA 0 KELL
530 POKE 161,CL%:    REM CIM-LO $A1-BE
540 POKE 162,CH%:    REM CIM-HI $A2-BE
550 SYS 53142:       REM ROM OLVASO RUTIN
560 POKE 4096+I,PEEK(2034)
570 RETURN
```

* Karakter JEL-mátrixbeli címének számítása:

```
cím=BLOKK*2048 + kód*8
```

2.3. 256 karakteres mód

Saját karakterkészlet használata esetén arra is van lehetőség, hogy az inverz karakterek használata helyett 256 különböző karaktert jelenítsünk meg. Ennek érdekében a TED 7. regiszterének 7. bitjét 1-re kell állítani:

```
POKE i,PEEK(i) OR 128                          i=65287 $ff07
```

2.4. Többszín mód

JELLEMZŐK: 25 sor * 40 jel
1 jel = 8 * 4pont (kvartett)

A többszín-módban karakterenként négy szín: a teljes képernyőre érvényes három háttérszín egyike, és a SZÍN-mátrixban karakterpozíciónként megadható szín közül választhatunk.

A többszín-mód és normál-mód karakterenként is váltogatható a SZÍN-mátrixban szereplő szín-kódtólfüggően:

-ha a szín-kód < 8, a megfelelő karakter-Pozíció 8*8bit (HI-RES) típusúként értelmeződik, színe 0-7.

-ha a szín-kód > 7, a pozíció 8*4 bit (MULTI) módban értelmeződik, a leíró 8 byte a JEL-mátrixban 8*4 bitpárra bomlik, és ezt értelmezi a TED-chip színkódként:

```
00: képernyő-szín(háttér#0)          65301 $ff15
01: háttér-szín#1                    65302 $ff16
10: háttér-szín#2                    65303 $ff17
11: előtér-színe(SZÍN-mátrix)
```

* Üzemmód beállítása:
```
POKE i,PEEK(i) OR 16                           i=65287 $ff07
```

* Üzemmód visszaállítása HI-RES-be (standard):
```
POKE i,PEEK(i) AND 239
```

Még: Többszín-mód és bővített háttérszín mód egyszerre nem használható!

2.5. Bővített háttérszín mód:

4-féle háttérszín között karakterpozíciónként lehet változatni. A jelek nyomtatásához a VIDEO-chip változatlanul a SZÍN-mátrixot használja. A 4-féle háttérszín kódját a KÉP-mátrix 6-7 bitjei tartalmazzák.

```
00: háttér reg#0                     65301 $ff15
01: háttér reg#1                     65302 $ff16
10: háttér reg#2                     65303 $ff17
11: háttér reg#3                     65304 $ff18
```

A karakterkódok tehát a következőképp állnak elő:

KÉPjel = karakterkód + 64*színregiszám

(KORLÁTOZÁS: Csak a JEL-mátrix BLOKKjainak első 64 karaktere használható.)

- Üzemmód beállítása:
```
POKE J,PEEK(J) OR 64                           J=65286 $ff06
```

- Üzemmód megszüntetése:
```
POKE J,PEEK(J) AND 191
```

3. NAGYFELBONTÁSÚ GRAFIKA

BASIC-ből való kényelmes kezelését számos grafikus utasítás könnyíti meg (BASIC fejezet, 10.2 pont).

3.1. Standard mód

```
                                                          cím
                                                        dec   hex

  *Alapadatok: 200 sor * 320 oszlop
               (25 sáv * 40 oktett)

     *BIT-mátrix: 8000 byte (1bit/pont)
          Elhelyezése alapértelmezésben:                 bbase=8192 $2000

     *SZÍN-mátrix: VIDEO-mátrix második Kbyte-ja         cbase=7168 $1c00
          1000 ($3e8) byte a háttér és jel szí-
          nének beállítására (1 byte / 8*8 bit).
          A SZÍN-mátrixban levő szín-kódok egy-
          egy 8*8-as pontnégyzetre érvényesülnek
                    bit 3-0  háttér színe
                    bit 7-4  pont színe

     *FÉNYERŐ-mátrix  VIDEO-mátrix első Kbyte-ja         fbase=6144 $1800
                    bit 2-0: adat fényerő
                    bit 6-4: háttér fényerő


  *Üzemmód használata


     *Üzemmód beállítása

          POKE j,PEEK(j) OR 32                              j=65286 $ff06


     *BIT-mátrix kijelölése

          POKE n,(PEEK(n) AND 199) OR g       bit 5-3       n=65298 $ff12
        ahol g=bbase/1024   alapért.: g=8


     *VIDEO-mátrix kijelölése

          POKE m,(PEEK(m) AND 7) OR e         bit 7-3       m=65300 $ff14
        ahol e=fbase/256    alapért.: e=16
        Megj: a fenti utasítás hatástalan,
              helyette a pszeudoregiszter
              használandó
          POKE f,fbase/256                                  f=2043  $07fb


     *BIT-mátrix törlése

          FOR I=bbase TO bbase+7999
          POKE I,0
          NEXT


     *Háttér-szín és fényerő beállítása

          FOR I=0 TO 999
          POKE cbase+I,16*pontszín#+háttérszín#
          POKE fbase+I,16*háttérfényerő+pontfényerő
          NEXT


     *BIT-térkép pontjainak címszámítása

        Leképezés  a kép 25sáv*40oktett-ből áll
                                                  0     x    319
          0 sáv 0 sor   0,  8, 16,....312 0+----------->
                1 sor   1,  9, 17,....313  !
                  .     .   .   .       .  !
                7 sor   7, 15, 23,....319 y!
          1 sáv 0 sor 320,328,336,....632  !
                  .     .   .   .       .  V
                                stb       199

          sáv#=INT(y/8)
          okt#=INT(x/8)
          sor#=y AND 7
          bit#=7-x AND 7
          byte#=bbase + sáv#*320 + okt#*8 + sor#
```

*Pont kigyújtása:

```
POKE byte#,PEEK(byte#) OR 2↑bit#
```

*Pont kioltása:

```
POKE byte#,PEEK(byte#)-2↑bit#
```

*Üzemmódból kilépés:

```
POKE J,PEEK(J) AND 223                          J=65286 $ff06
```

3.2. Többszín mód

*Alapadatok: 200 sor * 160 PontPár
(25 sáv * 40 kvartett)

*VIDEO-mátrix: mint standard-módban, de a színkódok hozzárendelése más,lsd. alább

*BIT-mátrix: 8000 byte (2 bit/PontPár)
A bitPárok azt jelölik ki, hogy az alábbi 4 lehetőség közül melyiket választjuk:

00 - háttérszín#0 65301 $ff15
01 - SZÍN-mátr. 4-7 bit fényerő matrix bit 2-0
10 - SZÍN-mátr. 0-3 bit fényerő mátrix bit 6-4
11 - háttérszín#1 65302 $ff16

*Üzemmód beállítása:

```
POKE J,PEEK(J) OR 32   HI-RES mód          J=65286 $ff06
POKE i,PEEK(i) OR 16   MULTI-mód           i=65287 $ff07
```

*Üzemmódból kilépés:

```
POKE J,PEEK(J) AND 223   HI-RES módból
POKE i,PEEK(i) AND 239   MULTI módból
```

4. EGYÉB GRAFIKUS LEHETŐSÉGEK:

4.1. SPRITE-ok (MOB-ok)

A C-16 nem tartalmaz lehetőséget valódi (a kép többi részétől és üzemmódjától független) sprite-ok megjelenítésére. Csak korlátozottan helyettesíti ezt a software úton megvalósított, képernyőrész eltárolását és a képernyő másik helyére való bemásolását, vagy transzformációját lehetővé tevő SSHAPE és GSHAPE BASIC utasítás (lsd. 16. old).

4.2. Scrolling(görgetés):

4.2.1. Karakter szintű scroll (karakteres képernyőn):

BASIC PRINT utasítással, a megfelelő escape szekvencia kiküldésével érhető el

4.2.2. Finom (Pontszintű)-scroll:

A képernyő görgetése folyamatosan végezhető a belépő oldal feltöltésének és a léptetésnek ciklikus ismétlésével. A léptetés a megfelelő (vízszintes vagy függőleges) scroll-regiszter 0-2 bitjein reprezentált (1 sor/oszlopnyi) eltolási érték megfelelő irányba való léptetésével valósítható meg. Az új (egy oszlopnyi/sávnyi, azaz 8 Pontnyi szélességű) adattal való feltöltés megfelelő sebességgel gépi-kódú rutinnal végezhető, de a BASIC PRINT utasítás is használható.
Ha egyszerre több képPont utasítás is használható. Ha egyszerre több képPont szélességű képrészletet (Pl. 8 Pont szélességű karaktereket) léptetünk be, a belépő karakter elfedését szolgálja a 24 ill. 38 karakteres üzemmód.

* Vízszintes mozgatás

* Átmenet 38 oszlopos üzemmódba:

```
POKE i,PEEK(i) AND 247                    i=65287 $ff07
```

Visszalépés 40 oszlopos módba:

```
POKE i,PEEK(i) OR 8
```

* Mozgatás irányának beállítása

```
POKE i,PEEK(i)  AND 248      (jobbra)
POKE i,(PEEK(i) AND 248)+7   (balra)
```

* Mozgatás: (például balra)

```
1.ide jön a feltöltést végző kód
FOR S=6 TO 0 STEP -1          (balra)
POKE i,(PEEK(i) AND 248)+S
NEXT
POKE i,(PEEK(i) AND 248)+7
2.vissza a feltöltésre
```

Rövid példa jobbra mozgatásra:

```
10 i=65287
20 Poke i,Peek(i) and 287
30 Poke i,Peek(i) and 248
40 Poke 3072,0: rem feltöltés
50 for s=1 to 7
60 Poke i,(Peek(i) and 248) + s
70 for j=3110 to 3072 step -1
80 Poke j+1,Peek(J)
90 next j
95 go to 30
```

* Függőleges mozgatás:

* Átmenet 24 soros üzemmódba:

```
POKE j,PEEK(j) AND 247                    j=65286 $ff06
```

Visszalépés 25 soros módba:

```
POKE j,PEEK(j) OR 8
```

* Mozgatás irányának beállítása:

```
POKE j,PEEK(j)  AND 248      (lefelé)
POKE j,(PEEK(j) AND 248)+7   (felfelé)
```

* Görgetés (például felfelé)

```
1.utolsó(25-ik) sor feltöltése
FOR S=6 TO 0 step -1
POKE j,(PEEK(j) AND 248)+S
NEXT
POKE j,(PEEK(j) AND 248)+7
2.vissza a feltöltésre
```

4.3. Képernyő elsötétítése:

A VIDEO-chip működése felfüggeszthető, ilyenkor a programok végrehajtása felgyorsul. A képernyő információ nem vész el.

*Kikapcsolás:

```
POKE j,PEEK(j) AND 239                    j=65286 $ff06
```

*Visszakapcsolás

```
POKE j,PEEK(j) OR 16
```

4.4. TED leállítása:

A tárfrissítés kivételével a TED valamennyi funkcióJa kikapcsolható.

*Kikapcsolás:

```
POKE i,PEEK(i) OR 32                    i=65287 $ff07
```

*Visszakapcsolás:

```
POKE i,PEEK(i) AND 223
```

4.5. NTSC szabványú videojel előállítása:

*NTSC jel:

```
POKE i,PEEK(i) OR 64                    i=65287 $ff07
```

*PAL jel:

```
POKE i,PEEK(i) AND 191
```

4.6. Színárnyalatok

A C16 újdonsága a C64-hez képest, hogy a 16 alapszínen belül (a fekete kivételével) 8-8 különböző színárnyalat használható, melyek kiválasztása a megfelelő színregiszter (vagy SZÍN-mátrix cella) 6-4 bitjein tárolható fényerő-érték (0-7) beállításával történik. A karakteres módban használható szín-billentyűk segítségével történő beállítás esetén alapértelmezés a legnagyobb fényerő (7), míg a szokott telítettségű színek a 4-5 fényerő-értékkel érhetők el.

Ez BASIC-ből a COLOR utasítással

```
COLOR színreg#,szín#,fényerő
   ahol:
      szín# : 1-16 (színkód+1)
      színreg# : 0-4
```

vagy POKE utasítással:

```
POKE cím,színkód+16*fényerő
```

állítható be.

4.7. Villogás

A színregiszterek legmagasabb helyértékű bitjének 1-re állítása (128=$80 hozzá OR-olása a regiszter tartalmához) a megfelelő terület villogását eredményezi, a bit kinullázása a villogást megszünteti.
Karakteres üzemmódban adott szöveg villogását a PRINT utasításból kiadott "FLASH ON", "FLASH OFF" karakterek közé való zárásával érhetjül el.

A villogás bekapcsolása egyéb esetekben:

```
POKE cím,PEEK(cím) OR 128
```

ill. kikapcsolása:

```
POKE cím,PEEK(cím) AND 127
```

4.8. Speciális képernyő-kezelés

* Raszter-regiszter

A képernyővel kapcsolatos tevékenységek szinkronizálására használható fel. A regiszter tartalma a kép-generálás pillanatnyi állapotát mutatja, nevezetesen az elektronsugár által pillanatnyilag rajzolt képernyő-sor sorszámát (0-311). Ha a kívánt változtatást akkor eszközöljük a képernyőn, amikor a raszter a nem látható területen van (regiszter tartalma nem 51 és 251 közé esik), elkerülhetjük a káros villódzást.

A regiszter címe: 65291 65291 $ff0b
A legmagasabb helyértékű bit(8) a 65290 0.-bitje. 65290 $ff0a

A raszter-regiszterbe előzetesen beírt szám elérése esetén a VIDEO chip megszakítást ad, ha a Program-megszakítást engedélyező regiszterben a megfelelő bit ON-ba(1) van állítva. 65290 $ff0a

* Program-megszakító állapot-regiszter 65289 $ff09

Külső megszakításokat jelzi. Interrupt bekövetkezésekor a megfelelő bit ON-ba áll, feltéve hogy az engedélyező regiszter megfelelő bitje ON. A már lekézelt interrupt-flag törlése a kérdéses bit újbóli ON-ba állításával lehetséges.

Bitjei:

1 - raszter-interrupt
2 - fényceruza negatív átmenete
3 - timer#1
4 - timer#2
6 - timer#3
7 - IRQ (bármely engedélyezett interrupt)

* Megszakítást engedélyező regiszter: 65290 $ff0a

Külső megszakítás engedélyezésére szolgál. Bitjei az állapot-regiszter bitjeinek felelnek meg. Az adott típusú megszakítás csak akkor jöhet be, ha az engedélyező regiszter megfelelő bitje ON (1).

Az interrupt lekezelésével a képernyő akár több, különböző üzemmódú részre osztható (felhasználására rövid példa a FÜGGELÉK J.4 pontjában található).

1. GÉPI-KÓDÚ UTASÍTÁSOK

Standard utasítások

típ	mnemo	hexakód	flag	címtíp	féle	szemantika
t	LDA	A0+a1	NZ	a1	8	M -> A
ö	LDX	A0+b1	NZ	b1	5	M -> X
l	LDY	A0+c1	NZ	c1	5	M -> Y
t	PLA	68	NZ	impl	1	stacktop -> A
	PLP	28	rest	impl	1	stacktop -> P (státusz-reg)
t	STA	80+a2	-	a2	7	A -> M
á	STX	80+b4	-	b4	3	X -> M
r	STY	80+c4	-	c4	3	Y -> M
o	PHA	48	-	impl	1	A -> stacktop
l	PHP	08	-	impl	1	P -> stacktop
c	TAX	AA	NZ	impl	1	A -> X
s	TAY	A8	NZ	impl	1	A -> Y
e	TSX	BA	NZ	impl	1	S -> X
r	TXA	8A	NZ	impl	1	X -> A
e	TXS	9A	-	impl	1	X -> S (stack-Pointer)
	TYA	98	NZ	impl	1	Y -> A
	BCS	B0	-	rel	1	ugrás ha carry ON (A>M)
v	BCC	90	-	rel	1	OFF (A<M)
e						
z	BEQ	F0	-	rel	1	ugrás ha zéro ON (A=M)
é	BNE	D0	-	rel	1	OFF (A<>M)
r						
l	BMI	30	-	rel	1	ugrás ha negative ON
é	BPL	10	-	rel	1	OFF
s						
	BVS	70	-	rel	1	ugrás ha overflow ON
á	BVC	50	-	rel	1	OFF
t						
a	BRK	00	BI	impl	1	force break, PC+2 -> stacktop
d						P -> stacktop ,transfer to OPSYS
á						
s	JMP	4C+d	-	d	2	jump: (PC+1)->PCL,(PC+2)->PCH
o	JSR	20	-	abs	1	jump and save return addr.in stck
k						
	RTI	40	rest	impl	1	visszatérés interrupt-ból
						stacktop -> P, stacktop - PC
	RTS	60	-	impl	1	visszatérés szubrutin-ból
						stacktop -> PC, PC+1 -> PC

A táblázat folytatását és a jelmagyarázatot lsd. a túloldalon!

Lehetséges címzési módok:

operandus forma	additív tag az adott címzési típusnál																utasítás hossz
	a1	a2	b1	b2	b3	b4	c1	c2	c3	c4	d	e1	e2	e3	f	
accum	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	1
immed	9	-	2	-	-	-	0	0	-	-	-	B	-	-	-	2
zpage	5	5	6	6	6	6	4	4	4	4	-	7	7	7	7	2
zpage,X	15	15	-	16	16	-	14	-	-	14	-	17	17	17	-	2
zpage,Y	-	-	16	-	-	16	-	-	-	-	-	-	-	-	17	2
absolut	D	D	E	E	E	E	C	C	C	C	0	E	E	E	E	3
abs,X	1D	1D	-	1E	1E	-	1C	-	-	-	-	1F	-	1F	-	3
abs,Y	19	19	1E	-	-	-	-	-	-	-	-	1B	-	1B	1F	3
(zpg,X)	1	1	-	-	-	-	-	-	-	-	-	3	3	3	3	2
(zpg),Y	11	11	-	-	-	-	-	-	-	-	-	13	-	13	13	2
(indir)	-	-	-	-	-	-	-	-	-	-	20	-	-	-	-	3

típ	mnemo	hexakód	flag	címtíp	féle	szemantika
lo	AND	20+a1	NZ	a1	8	M AND A -> A
gi	ORA	00+a1	NZ	a1	8	M OR A -> A
kai	EOR	40+a1	NZ	a1	8	M XOR A -> A
ha	CMP	C0+a1	NZC	a1	8	A - M
son	CPX	E0+c2	NZC	c2	3	X - M
lí	CPY	C0+c2	NZC	c2	3	Y - M
tás	BIT	20+c3	NZ V	c3	2	A AND M, M7 -> N, M6 -> V
la	ADC	60+a1	NZCV	a1	8	A+M+C -> A,C
r	INC	E0+b3	NZ	b3	4	M+1 -> M
i	INX	E8	NZ	impl	1	X+1 -> X
t	INY	C8	NZ	impl	1	Y+1 -> Y
le	SBC	E0+a1	NZCV	a1	8	A-M-(NOT C) -> A,C
t	DEC	C0+b3	NZ	b3	4	M-1 -> M
i	DEX	CA	NZ	impl	1	X-1 -> X
k	DEY	88	NZ	impl	1	Y-1 -> Y
sh	ASL	00+b2	NZC	b2	5	C <- 76543210 <- 0
i	LSR	40+b2	NZC	b2	5	0 -> 76543210 -> C
f	ROL	20+b2	NZC	b2	5	C <- 76543210 <- C
t	ROR	60+b2	NZC	b2	5	C -> 76543210 -> C
	CLC	18	C	-	1	clear carry flag -> 0
e	CLD	D8	D	-	1	clear decimal mode flag-> 0
g	CLI	58	I	-	1	clear interrupt disable-> 0
y	CLV	B8	V	-	1	clear overflow flag -> 0
é						
b	NOP	EA	-	-	1	no operation
	SEC	38	C	-	1	set carry flag -> 1
	SED	F8	D	-	1	set decimal mode -> 1
	SEI	78	I	-	1	set interrupt disable -> 1

Lehetséges címzési módok:

operan- dus forma	additív tag az adott címzési típusnál																lutasí tás	
	a1	a2	b1	b2	b3	b4	c1	c2	c3	c4	d	e1	e2	e3	f	hossz		
akcum	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	1		
immed	9	-	2	-	-	-	0	0	-	-	-	B	-	-	-	2		
zPage	5	5	6	6	6	6	4	4	4	4	-	7	7	7	7	2		
zPage,X	15	15	-	16	16	-	14	-	-	14	-	17	17	17	-	2		
zPage,Y	-	-	16	-	-	16	-	-	-	-	-	-	-	-	17	2		
absolut	D	D	E	E	E	E	C	C	C	C	0	E	E	E	E	3		
abs,X	1D	1D	-	1E	1E	-	1C	-	-	-	-	1F	-	1F	-	3		
abs,Y	19	19	1E	-	-	-	-	-	-	-	-	1B	-	1B	1F	3		
(zPg,X)	1	1	-	-	-	-	-	-	-	-	-	3	3	3	3	2		
(zPg),Y	11	11	-	-	-	-	-	-	-	-	-	13	-	13	13	2		
(indir)	-	-	-	-	-	-	-	-	-	-	20	-	-	-	-	3		

Jelmagyarázat:

N - negatív indikátor
Z - zéró indikátor
C - carry indikátor/bit
V - overflow indikátor
D - decimális-mód flag
I - interrupt letiltó flag
B - break indikátor

P - státusz-regiszter
S - stack-Pointer regiszter
A - akkumulátor regiszter
X - X index-regiszter
Y - Y index-regiszter
PC - Program-számláló
M - memória-cella v. operandus

hexakód: műveleti kód [+ +imzési módtól függő additív tag]
flag: az utasítás által módosított indikátorok
címtíp: a megengedett címzési típus(ok) gyűjtőjele
féle különböző címzési lehetőségek száma
rest: előző érték visszatöltése (restore)
impl: az operandus a műveletből következik(implied)
imm.: az operandus a műveleti kód után következő byte(immediate)
rel.: a címrész az utasításszámláló Pillanatnyi értékéhez viszonyított relatív offszet (-128 - +127)
0-as címrész a következő utasításra mutat!

2. PÁRHUZAMOS TÁBLÁZATOK

Standard ASCII kódtábla
(Jelmagyarázatot lsd. D. FÜGGELÉK-ben)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	sPc	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	bsl	]	↑	←
6	sPc	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	bkP	|	jkP	ház	DEL

Hexadecimális-decimális konverziós tábla

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

A gépi utasítások kódjai és címzés/operandus-típusuk:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BRK	ORA		ASO		ORA	ASL		PHP	ORA	ASL			ORA	ASL	
		9		9		3	3			2	1			6	6	
1	BPL	ORA		ASO		ORA	ASL		CLC	ORA	NOP			ORA	ASL	
	c	a		a		4	4			7				8	8	
2	JSR	AND		RLA	BIT	AND	ROL		PLP	AND	ROL		BIT	AND	ROL	
	6	9		9	3	3	3			2	1		6	6	6	
3	BMI	AND		RLA		AND	ROL		SEC	AND	NOP			AND	ROL	
	c	a		a		4	4			7				8	8	
4	RTI	EOR		LSE		EOR	LSR		PHA	EOR	LSR		JMP	EOR	LSR	
		9		9		3	3			2	1		6	6	6	
5	BVC	EOR		LSE		EOR	LSR		CLI	EOR	NOP			EOR	LSR	
	c	a		a		4	4			7				8	8	
6	RTS	ADC		RRA		ADC	ROR		PLA	ADC	ROR		JMP	ADC	ROR	
		9		9		3	3			2	1		b	6	6	
7	BVS	ADC		RRA		ADC	ROR		SEI	ADC	NOP			ADC	ROR	
	c	a		a		4	4			7				8	8	
8		STA			STY	STA	STX		DEY		TXA		STY	STA	STX	
		9			3	3	3						6	6	6	
9	BCC	STA			STY	STA	STX		TYA	STA	TXS			STA		
	c	a			4	4	5			7				8		
A	LDY	LDA	LDX	LAX	LDY	LDA	LDX		TAY	LDA	TAX		LDY	LDA	LDX	
	2	9	2	9	3	3	3			2			6	6	6	
B	BCS	LDA		LAX	LDY	LDA	LDX		CLV	LDA	TSX		LDY	LDA	LDX	
	c	a		a	4	4	5			7			8	8	7	
C	CPY	CMP		DCM	CPY	CMP	DEC		INY	CMP	DEX		CPY	CMP	DEC	
	2	9		9	3	3	3			2			6	6	6	
D	BNE	CMP		DCM		CMP	DEC		CLD	CMP	NOP			CMP	DEC	
	c	a		a		4	4			7				8	8	
E	CPX	SBC		INS	CPX	SBC	INC		INX	SBC	NOP		CPX	SBC	INC	
	2	9		9	3	3	3			2			6	6	6	
F	BEQ	SBC		INS		SBC	INC		SED	SBC	NOP			SBC	INC	
	c	a		a		4	4			7				8	8	

Címzési típus (operandus típus):

1: akkumulátor 2: immediate 3: zeroPage 4: zeroPage,X
5: zeroPage,Y 6: abszolút 7: abszolút,Y 8: abszolút,X
9: (zeroPage,X) a: (zeroPage),Y b: indirekt c: relatív

3. A KERNAL

3.1. KERNAL táblázatok

tartalom	név	hex	dec
aktív logikai file# tábla	LAT	$509-$512	1289-1298
készülékszám-tábla	FAT	$513-$51c	1299-1308
másodlagos-cím (csatorna#) tábla	SAT	$51d-$526	1309-1318
OS memória-kezdőcím Ptr	MEMSTR	$531,$532	1329,1330
OS memória felső-határ	MEMSIZ	$533,$534	1331,1332
hardware IRQ interrupt vektor	CINV	$314,$315	788,789
BRK utasítás interrupt vektor	CBINV	$316,$317	790,791
nem maszkolható interrupt vektor	NMINV	$312,$313	786,787
ST I/O státusz-szó	STATUS	$090	144
real-time jiffy-clock	TIME	$a3- $a5	163-165
MONITOR ST, A, X, Y regiszterek		$554-$557	1364-1367
MONITOR SP-reg		$558	1368
MONITOR PC		$552,$553	1362,1363

8301 processzor I/O Port-ja $0000-$0001 0-1

$0	IN	IN	OUT	IN	OUT	OUT	OUT	OUT
$1	IEC-bus DATA	IEC-bus CLOCK		tape sense	tape motor	IEC-bus ATN	IEC-bus CLOCK	IEC-bus DATA

A TED 7360 regiszterei:

```
$fd00-$fd03 - RS232 I/O
$fd10 - szalagegység

                bit 2 - 0 : PLAY ON
                        1 :      OFF

$fd30 - 6529 keyboard-Port (oszlop kijelölése)
$fdd0 - BANK-Port

        fdd0: ROM-enable
              bit 0-1  -  00 : BASIC
                          01 : -
                          10 : C1 low
                          11 : C2 low
                    2-3  -  00 : KERNAL
                          01 : -
                          10 : C1 high
                          11 : C2 high

$fec0-$fec5,$fef0-$fef5 - Paralel IEC I/O
$ff00-$ff1f - hang/video/timer/interrupt/keyboard regiszterek
              (lsd. a ZENE és a GRAFIKA c. fejezeteket)
$ff3e-$ff3f - ROM enable/disable
```

3.2. KERNAL rutinok

Jelmagyarázat:

```
regiszter:tartalom> - input paraméter(regiszterben),
                      hívás előtt betöltendő
>regiszter:tartalom - visszaadott érték
mscím - másodlagos cím v. parancskód
lf#   - (logikai) file#
kész# - készülékszám (device number)
C - carry-bit
S - hibastátusz, READST-vel lekérdezhető
E - hibastátusz az A regiszterben
Param.reg. - paraméter átadására felhasznált regiszter
előzmény - rutinhívás elkészítéséhez szükséges rutinok
[előzmény] - csak első alkalommal szükséges
használt reg/err - rutinban használt regiszterek és hibastátusz
                   visszaadásának módja
```

feladat	név	cím	Param. reg.	előz-mény	használt reg/err
init VIC és KERNAL	CINT	ff81	-	-	A,X,Y
RAM tesztelése és inicializálása	RAMTAS	ff87	-	-	A,X,Y
RAM kezdőcímének beállítása ill. kiolvasása	MEMBOT	ff9c	C 0>,XY> C 1>,>XY	-	X,Y,C
RAM felső határának beáll. ill. kiolvasása	MEMTOP	ff99	C 0>,XY> C:1>,>XY	-	X,Y,C
rendszer RAM vektorok beáll. ill. kiolvasása (32byte) XY: user terület kezdőcíme	VECTOR	ff8d	C:0>,XY> C:1>,XY>	-	A,X,Y
rendszer-vektorok helyreáll.	RESTOR	ff8a	-	-	A,X,Y
kurzor Poz. beállítás ill. kiolvasás (X:sor Y:oszl.)	PLOT	fff0	C:0>,XY> C 1>,>XY	-	A,X,Y
képernyő sor és oszlopszám visszaadása (X:oszl Y:sor)	SCREEN	ffed	>X,>Y	-	X,Y
I/O memória báziscím lekérd	IOBASE	fff3	>XY addr	-	X,Y,E
I/O egységek inicializálása	IOINIT	ff84	-	-	A,X,Y
logikai file#,készülék# és másodl.cím beállítása A:file# X:készülék# Y:másodlagos cím/parancs (ha nincs,255)	SETLFS	ffba	A>,X>,Y>	-	
file-név beállítása A:név hossza X,Y:név címe (lo,hi)	SETNAM	ffbd	A>,XY>	-	
logikai file megnyitása	OPEN	ffc0	-	SETLFS SETNAM	A,X,Y S,E
csatornamegnyitás inputra	CHKIN	ffc6	X:lf#>	[OPEN]	A,X,E
csatornamegnyitás outputra	CHKOUT	ffc9	X lf#>	[OPEN]	A,X,E
karakter csatornán be	CHRIN	ffcf	>A:byte	[OPEN] [CHKIN]	A,X S,E
billentyűzet letapogatása érték a puffer-sorba	SCNKEY	ff9f	-	[IOINIT]	A,X,Y
karakter keyb.Pufferből vagy RS232 csatornáról	GETIN	ffe4	>A:byte	[OPEN] [CHKIN]	A,X,Y S
STOP-gomb leérzékelése ha volt STOP, Z-flag=1 ha nem, A:keyboard-scan utolsó sora	STOP	ffe1	>A, >Z	-	A,X
karakter csatornán ki	CHROUT	ffd2	A:byte>	[OPEN] [CHKOUT]	A,S,E
logikai file lezárása	CLOSE	ffc3	A:lf#>	-	A,X,Y,E
összes csatorna lezárása	CLALL	ffe7	-	-	A,X
I/O csatornák törlése (reset default csatorna)	CLRCHN	ffcc	-	-	A,X
ST státuszbyte olvasása	READST	ffb7	>A:byte	-	A

	eszköz soros buszon TALK	TALK	ffb4	A:kész#>	-	A,S
S	másodl.cím/parancs TALK módú eszköznek	TKSA	ff96	A:mscím>	TALK	A,S
O	adatbevitel soros buszról n (telj. IEEE handshake-kel)	ACPTR	ffa5	>A:byte	TALK TKSA	X,A,S
P	eszköz adatokat s.buszról folyamatosan vesye(CMD)	LISTEN	ffb1	A:kész#>	-	A,S
Ó	másodlagos cím LISTEN-hez	SECOND	ff93	A:mscím>	LISTEN	A,S
S	karakter soros buszra	CIOUT	ffa8	A:byte>	[LISTEN] [SECOND]	S
	IEEE busz-kártya timeout flag enable: bit#7=0	SETTMO	ffa2	A>	-	-
B	disable:bit#7=1					
	soros buszon összes eszköz	UNTLK	ffab	-	-	A,S
U	fűggessze fel TALK-ot					
	összes eszköz soros buszról	UNLSN	ffae	-	-	A,S
S	kapcsolódjon le					

S	memória kiírása log.filera	SAVE	ffd8	A>,XY>	SETLFS	A,X,Y
A	A->zpage ptr.->kezdőcím				SETNAM	E,S
V	X,Y -> végcím					
L	RAM töltése (LOAD) vagy	LOAD	ffd5	A:0>,XY>	SETLFS	A,X,Y
Ó	VERIFY logikai eszközről			A:1>	SETNAM	S,E
D	X,Y: kezdocím(lo,hi)					

S	rendszer-üzenet kiküldése	SETMSG	ff90	A:msg#>	-	A
	bit#7-ON:hibaüzenet					
Y	bit#6-ON:utasítás					
S	rendszer-óra beáll.(3byte)	SETTIM	ffdb	AXY>	-	
T						
	rendszer óra 3byte-os érté-	RDTIM	ffde	>AXY	-	A,X,Y
E	kének kiolvasása					
M	rendszer-óra update-je	UDTIM	ffea	-	-	A,X

	MONITOR meghivása	MONITR	ff52	-	-	
E						
	szöveg hozzárendelése funk-	ASGFUN	ff49	A:hossz>		
G	ciógombhoz			X:gomb#>		
	$22/$23:szöveg címe>					
Y						
	szöveg kiírása aktuális pe-	PRINIT	ff4f			
É	rifériára (szöveg a JSR					
	utasítás után közvetlenül,					
B	0-byte zárja)					

3.3. KERNAL státuszok:

Ha a KERNAL-rutinok végrehajtása során hiba következik be, a carry-flag ON-ba kerül és az A regiszter a hibakódot fogja tartalmazni. Lehetséges hibakódok:

0 - a rutin félbeszakítását a STOP gombbal kezdeményezték
1 - túl sok file van megnyitva
2 - a file már nyitva van
3 - a file nincs megnyitva
4 - a file-t nem találja
5 - az eszköz nincs jelen
6 - a file nem inPut-file
7 - a file nem outPut-file
8 - a file-név hiányzik
9 - illegális készülékszám

Egyes KERNAL-rutinok hiba esetén a STATUS-regiszterben (ST) adják vissza a hiba természetére vonatkozó információt, ami aztán a READST rutinnal olvasható ki:

kód	szalag	soros I/O	szalag verify/load
1		out-timeout	
2		in -timeout	
4	rövid-blokk		rövid-blokk
8	hosszú-blokk		hosszú-blokk
16	nem javítható olvasás-hiba		bármilyen eltérés
32	kontrolszumma hiba		kontrolszumma hiba
64	file-vége	EOI line	
-128	szalag-vége	egység nincs jelen	szalag-vége

4. GÉPI-KÓDÚ RUTINOK HASZNÁLATA BASIC PROGRAMOKBÓL

4.1. BASIC Programok kezelésével kapcsolatos fontosabb memóriacímek

BASIC Program kezdőcíme (Pointer)	TXTTAB	$2b,$2c	43,44
Load Programkezdet		$b2,$b3	178,179
Programvég		$b4,$b5	180,181
BASIC változók kezdőcíme (Ptr)	VARTAB	$2d,$2e	45,46
BASIC tömbök kezdőcíme (Ptr)	ARYTAB	$2f,$30	47,48
Szabad memória kezdőcíme (Ptr)	STREND	$31,$32	49,50
BASIC verem-mutató (Ptr)		$7c,$7d	124,125
String-tárolóterület alja (Ptr)	FRETOP	$33,$34	51,52
String segéd-Pointer	FRESPC	$35,$36	53,54
BASIC által használt legmagasabb cím	MEMSIZ	$37,$38	55,56
Kurrens BASIC-sor sorszáma	CURLIN	$39,$3a	57,58
BASIC-szöveg kurrens byte-ja (Ptr)	TXTPTR	$7a,$7b	122,123
Előző BASIC-sor sorszáma	OLDLIN	$259,$25a	601,602
CONT visszalépési sorszám (Ptr)	OLDTXT	$25b,$25c	603,604
Kurrens DATA-sor sorszáma	DATLIN	$3f,$40	63,64
Kurrens DATA-elem címe (Ptr)	DATPTR	$41,$42	65,66
Kurrens BASIC-változó neve	VARNAM	$45,$46	69,70
Kurrens BASIC-változó adat (Ptr)	VARPNT	$47,$48	71,72
Dimenziószám		$0b	11
INPUT - Pointer		$43,$44	67,68
String-Pointer		$50-$53	80-83
AUTO növekmény		$73,$74	115,116
FLAG-EK			
DIM-flag		$0c	12
Szám/szöveg flag ($ff:szöveg)		$0d	13
Egész/lebegőpontos flag ($80:egész)		$0e	14
Idézőjel-flag		$0f	15
STOP flag		$91	145
Parancs/Program mód flag		$81	129
HIBAKEZELÉS			
Hibás sor sorszáma		$4f0,$4f1	1264,1265
Hibakód		$4ef	1263
Hiba címe		$4f5,$4f6	1269,1270
TRAP-sor sorszáma		$4f2,$4f3	1266,1267
CIKLUS			
FOR/NEXT ciklusváltozó (Ptr)	FORPNT	$49,$4a	73,74
kurrens DO címe	DOPNT	$4f8,$4f9	1272,1273
DO-sor sorszáma		$4fa,$4fb	1274,1275
REPEAT-számláló		$541,$542	1345,1346
JMP USR		$500-$502	1280-1282
FN - Pointer		$4e,$4f	78,79
FN - flag		$10	16
JMP-utasítás fügvény végrehajtáshoz		$54-$56	84-86
GRAFIKA			
Grafikus-terület flag		$75	117
HI/MULTI flag		$83	131
Grafikus utasítás színregiszter-spec (0-4)		$84	132
Multicolor-színek		$85,$86	133,134
PAINT-mód flag		$8b	137

LOGIKAI I/Ó			
Default input készülék (0)	DFLTN	$98	152
Default output CMD készülék (3)	DFLTO	$99	153
Nyitva levő file-ok száma (index a file-táblához)	LDTND	$97	151
Kurrens logikai file#	LA	$ac	172
Kurrens file-név hossza	FNLEN	$ab	171
Kurrens másodlagos cím /Parancskód	SA	$ad	173
Kurrens készülék#	FA	$ae	174
Kurrens file-név (Ptr)	FNADR	$af,$b0	175,176
ST rendszerváltozó		$90	144
DS$ érvényesség flag		$79	121
DS$ címe (Ptr)		$7a,$7b	122,123

REGISZTEREK			
.A regiszter tárolóhely	SAREG	$7f2	2034
.X regiszter tárolóhely	SXREG	$7f3	2035
.Y regiszter tárolóhely	SYREG	$7f4	2036
.ST regiszter tárolóhely	STREG	$7f5	2037
Lebegőpontos akkumulátor#1 exponens	FACEXP	$61	97
mantissza	FACHO	$62-$65	98-101
előjel	FACSGN	$66	102
overflow-digit	BITS	$68	104
low-order(kerekítés)	FACOV	$70	112
Lebegőpontos akkumulátor#2 exponens	ARGEXP	$69	105
mantissza	ARGHO	$6a-$6d	106-109
előjel	ARGSGN	$6e	110
Előjel-összehasonlítás(FAC1 vs. FAC2) eredménye	ARISGN	$6f	111
Lebegőpontos akkumulátor#3 (FAC3)		$57-$5b	87-91
Lebegőpontos akkumulátor#4 (FAC4)		$5c-$60	92-96

4.2. Szabadon felhasználható területek

211-232	$d1-$e8	szabad nullás-lap (zero-page) helyek
636-684	$27c-$2ac	diszk Parancs-Puffer(ha nincs diszk)
758-767	$2f6-$2ff	nem használt terület
818-1010	$332-$3f2	szalag-Puffer (ha nincs szalagegység)
1015-1078	$3f7-$436	RS232-Puffer
1186-1188	$4a2-$4a4	nem használt terület
1368-1370	$558-$55a	nem használt terület
1525-1771	$5f5-$6eb	nem használt terület
2046-2047	$7fe-$7ff	nem használt terület

4.3. A BASIC interpreter és a KERNÁL főbb user-belépési pontjai

(vektorok : az indirekt címeket alsó('lo'),felső('hi') sorrendben tartalmazó bytePárok = Pointerek):

BASIC meleg-start	IMAIN	$300,$301	768,769
BASIC hibauzenet-kiírás (X: hibakód)	IERROR	$300,$301	768,769
BASIC-sor beolvasása		$302,$303	770,771
BASIC text kódolás (tokenizálás)	ICRNCH	$304,$305	772,773
BASIC LIST (token nyomtatás)	IOPLOP	$306,$307	774,775
BASIC text elemző (aritmetikai elem)	IEVAL	$30a,$30b	778,779
új BASIC-kód kezdése(char.dispatch)	IGONE	$308,$309	776,777
Felhasználói kód tokenizálás		$30c,$30d	780,781
Felhasználói token listázás		$30e,$30f	782,783
Felhasználói token végrehajtás		$310,$311	784,785
USR függvény ugró utasítás	USRPOK	$500-$502	1280-1282
OPEN rutin	IOPEN	$318,$319	792,793
CLOSE rutin	ICLOSE	$31a,$31b	794,795
CHKIN rutin	ICHKIN	$31c,$31d	796,797
CHKOUT rutin	ICKOUT	$31e,$31f	798,799
CLRCHN rutin	ICLRCH	$320,$321	800,801
CHRIN rutin	IBASIN	$322,$323	802,803
CHROUT rutin	IBSOUT	$324,$325	804,805
STOP rutin	ISTOP	$326,$327	806,807
GETIN rutin	IGETIN	$328,$329	808,809
CLALL rutin	ICLALL	$32a,$32b	810,811
LOAD rutin	ILOAD	$32e,$32f	814,815
SAVE rutin	ISAVE	$330,$331	816,817
Billentyű dekódolás vektor		$545,$546	1349,1350

A fenti vektorok felhasználhatók a BASIC utasítások feldolgozásának módosítására, ha tartalmukat úgy változtatjuk, hogy saját rutinunkra mutasson.

BASIC-szöveg kov.byte-jának beolv. az A regiszterbe	CHRGET	$473	1139
BASIC-szöveg byte-jának újraolvasása az A regiszterbe	CHRGOT	$479	1145

Példa a LIST módosítására a listázás a SHIFT billentyű lenyomására megáll, felengedve tovább folyik (SHIFT-LOCK-al rögzíthető):

```
shflag = $0543     ; ha 1, SHIFT lenyomva
slist  = $b6e      ; LIST token rutinja
*      = $05f5     ; betoltesi cim 1525
       pha         ; A mentése
wait   lda shflag  ; SHIFT lenyomva ?
       bne wait    ; igen, várunk
       pla         ; nem, A-t visszaállitjuk
       jmp slist   ; LIST mehet
```

A rutin aktivizálása (a vektor átállítása):

```
POKE 774,245  POKE 775,5
```

4.4. RAM és ROM olvasó rutinok:

A teljes (64 kbyte-ra bővíthető) RAM kihasználása érdekében a ROM-mal fedett területek olvasása előtt a BASIC-interpreter a ROM-ot kikapcsolja, és a 0-ás lapon tárolt vektorok felhasználásával a mögötte levő RAM-ot olvassa. A RAM-területen elhelyezett szubrutinok közvetlenül hívhatók. Hívás előtt az Y regiszterbe 0 töltendő (vagy a báziscímhez képesti offszet), a vektorba a báziscím, A-ba a vektor címe. Ez utóbbi elhagyható, ha a BASIC-ROM-beli belépési ponton keresztül hívjuk meg őket. Az eredményt az A-regiszterben kapjuk.

ROM-belépési pont	RAM-cím	vektor címe	Megjegyzés
$8155	$0494	$43	A betöltendő
$8159	$0494	$4e	,,
$815d	$0494	$14	,,
$8161	$0494	$47	,,
$8165	$0494	$4e	,,
$8169	$0494	$5c	,,
$816d	$0494	$5f	,,
$8171	$0494	$3d	,,
$8175	$0494	$57	,,
$8179	$0494	$59	,,
$817d	$0494	$62	,,
$8181	$0494	$50	,,
$8185	$0494	$6c	,,
$8189	$0494	$5a	,,
.	$04a5	$3b	fix vektoros
	$04b0	$22	,,
	$04bb	$24	,,
	$04c6	$6f	,,
	$04d1	$5f	,,
	$04dc	$64	,,

A ROM olvasása viszont BASIC-ben okoz problémát, mert a PEEK függvény (C64-beli párjától eltérően) mindig a RAM-ot olvassa. A ROM olvasására szolgáló rutin paraméterezése hasonló: $a1, $a2-be a báziscím, Y-ba a báziscímhez képesti offszet töltendő; az eredményt az A-ban kapjuk. Hívása BASIC-ből (lsd. GRAFIKA, 2.2 pont):

```
SYS 53142
```

Gépi kódú rutinból:

```
JSR $cf96
```

4.5. Rutinhívás és paraméterátadás BASIC programból

Gépi-kódú rutinok hívása a SYS utasítással történhet BASIC programokból. A SYS utasítás szintaxisa alapállapotban:

```
SYS cikfejezés
```

Ilyenkor rendszerint felmerül egy vagy több paraméter átadásának problémája is:

4.5.1. A POKE és PEEK használata

Paraméterek átadhatók gépi-kódú rutinoknak a POKE utasítás felhasználásával, pl. a $7f2 címen kezdődő regiszter-tároló területen, a BASIC interpreter ugyanis az ST, A, X, Y regisztereket a vezérlésátadás előtt innen automatikusan betölti, illetve visszatérés előtt ide kitárolja (Megj: $14,15 tartalmát elrontja).

Példa: Gépi-kódú rutin a billentyű-mátrix felhasználására, a leütött billentyű letapogatására (lehetővé teszi egyszerre lenyomott több billentyű azonosítását is!):

```
*=$033d
sei             interruptot letiltjuk
sta $fd30       maszkot beírjuk a "row-select reg.be
sta $ff08       ide is beírjuk
lda $ff08       kiolvassuk a "column read" flag-eket
cli             interruptot felszabadítjuk
rts             visszatérünk a hívás helyére
```

A rutint hívó BASIC-program:

```
1 INPUT "SOR=";S
2 POKE 2034,255-21S:      REM MASZK KITAROLASA SAREG-BE
3 SYS 829:                REM SZUBRUTIN MEGHIVASA
4 PRINT S,255-PEEK(2034): REM FLAG-BYTE SAREG-BöL
5 GOTO 2
```

4.5.2. [illegible] kezelése gépi-kódú rutinokból (Paraméterátadás,stb)

A [illegible] interpreter felsorolásra kerülő rutinjai jelentős segítséget nyújthatnak BASIC programok gépi-kódú kiegészítéséhez (Pl. a SYS utasítás felhasználásához nemcsak a saját fejlesztésű rutin meghívására, hanem szimbolikus - sőt kifejezést tartalmazó - paraméterek átadására is)

feladat	név	cím	előzmény	Paraméterek
Pozicionálás BASIC sorban, [illegible]következő vesszőig	CHKOMA	$9491	-	-
Pozicionálás BASIC sorban, [illegible]következő) -ig	CHRPRN	$948b	-	-
Pozicionálás BASIC sorban, következő (-ig	CHLPRN	$948e	-	-
Pozicionálás BASIC sorban, adott karakterig	CHDELM	$9493	-	A:keresett karakter>
nkif Paraméter kiértékelése és átvitele FAC-ba	FRMNUM	$9314	CHKOMA	>FAC:nParam
bkif Paraméter kiértékelése és átvitele -be	GTBYT	$9d84	CHKOMA	>x:bParam
sParam beolvasása és leírójának átadása	GETSTR	$9d67	CHKOMA	>y:hossz >$22,23:cím
ckif,bkif kiértékelése és átadása $14,15 ill. x-ben	GETCOR	$9dd2	CHKOMA	>$14,15:ckif >x:bkif
FAC konvertálása címmé és átadása $14,15-be	GETADR	$9de1	FRMNUM	>$14,15:ckif
"illegal quantity" kiírása, futás megszakítása	ILLQNT	$991c	-	-
cr kiírása a képernyő aktuális Pozíciójába	CPLOT	$e001	-	A:karakter> $53b:szín >
FAC konverziója és kiírása aktuális Poz.-ba		$a46f	-	-
A/X két-byte-os egész kiiratása képernyőre		$a45f	-	A:hi-byte> X:lo-byte>

4.5.3. Lebegőpontos műveletek gépi-kódban

Mivel közvetlen lebegőpontos gépi kódú utasítások nincsenek, ezek a műveletek csak igen sok lépésben végezhetők el. A BASIC interpreter erre szolgáló rutinjainak felhasználása azonban ezt is kényelmessé teszi.

A hatékony programszervezés érdekében kétféle rutinkészlet is van:

a) a műveleteket a lebegőpontos akkumulátor#1 (FAC, $61-65) és egy A/Y (helyenként X/Y) regiszterpárral címzett 5 byte-os memóriarekesz között végző, illetve
b) a műveleteket a FAC és a lebegőpontos akkumulátor#2 (ARG, [illegible]) között végző

szubrutinok (belépési pontok).

Az operandusok előkészítése a rutinhívás előtt a programozó feladata.

feladat		név	cím	Paraméterek
FAC = M(A,Y)	(tölt)	LDFAC1	$a21f	A,Y:memóriacím >
M(X,Y) = FAC	(tárol)	STFAC1	$a259	X,Y:memóriacím >
ARG = M(A,Y)	(RAM-ból)	LDFAC2	$a107	A,Y:memóriacím >
ARG = M(A,Y)	(ROM-ból)	LDFAC2	$a0dc	A,Y:memóriacím >
FAC = M(A,Y)+FAC	(+)	ADDMF1	$9e9b	A,Y:oP#1cím >
FAC = M(A,Y)*FAC	(*)	MPYMF1	$a078	A,Y:oP#1cím >
FAC = M(A,Y)/FAC	(/)	DIVMF1	$a194	A,Y:oP#1cím >
FAC = ARG↑M(A,Y)	(↑)	EXPF1M	$a5eb	A,Y:oP#2cím >
FAC = ARG + FAC	(+)	ADF2F1	$9e9e	A:$61-tartalma >
FAC = ARG * FAC	(*)	MPF2F1	$a078	A:$61-tartalma >
FAC = 10 * FAC	(*10)		$a162	-
FAC = ARG - FAC	(-)	SBF2F1	$9e87	-
FAC = ARG / FAC	(/)	DVF2F1	$a197	A:$61-tartalma >
FAC = FAC / 10	(/10)		$a193	-
FAC = ARG ↑ FAC	(↑)	EXF2F1	$a5ee	A:$61-tartalma >
FAC = -FAC	(neg)	NEGF1	$a627	-
FAC = ARG	(másol)	CPYF2	$a281	-
ARG = FAC	(másol)	CPYF1	$a291	-
FAC = ABS(FAC)	(abs)	ABSFAC	$a2dd	-
FAC = ROUND(FAC)	(kerekít)	RNDF1	$a2a0	-
ARG = ROUND(FAC)	(kerekít)	CPRNF1	$a291	-
FAC = NORML(FAC)	(normál)	NORMF1	$9f32	-
M(98-101) = INT(FAC) (4 byte-os egésszé konv.)		LINTF1	$a327	-
M(100-101) = INT(FAC) (2 byte-os egésszé konv.)		INTF1	$9886	-
A,Y = INT(FAC)	(konv.)	AYINT	$9871	> A,Y:integer
FAC = FLOAT(A,Y)	(konv.)	FLTAY	$9a92	A,Y integer >
FAC hasonlítása M(A,Y)-hoz		CMPMF1	$a2e0	A,Y:oP#cím > > A:eredmény 00 - M=FAC 01 - M<FAC FF - M>FAC
FAC előjele		SGNFAC	$a2b0	> A:eredmény 01 - Pozitív ff - negatív

4.5.4. Példák a fenti rutinok felhasználására:

Lsd. J.5. FÜGGELÉK.

5. A MEMÓRIA KEZELÉSE

A C-16 által kezelhető 64 Kbyte-os címtartományban (alaphelyzetben) 16 Kbyte írható-olvasható RAM és 32 Kbyte (olvasható) ROM memória helyezkedik el oly módon, hogy a ROM területek (bővítés esetén) a RAM egyes zónáit fedik át. Ez azzal jár, hogy ezekben a címtartományokban az olvasás alaphelyzetben a ROM-ból, az írás pedig a RAM-ba történik. Az ebből eredő esetleges problémákat úgy kerülik el, hogy mind a BASIC-interpreter, mind a KERNAL speciális RAM és ROM-olvasó rutinokat tartalmaz (lsd. ROM/RAM térkép, K. FÜGGELÉK).

A teljes címtartomány 7 fő zónára oszlik:

zóna	cím hex	cím dec	méret	alapállapot	átkapcsolható
#1	$0000-$3fff	0-16383	16K	RAM	
#2	$4000-$7fff	16384-32767	16K	üres	külső-ROM/RAM
#3	$8000-$cfff	32768-53247	20K	BASIC-ROM	külső-ROM/RAM
#4	$d000-$d7ff	53248-55295	2K	karakter-ROM	külső-ROM/RAM
#5	$d800-$fcff	55296-64767	9K	KERNAL-ROM	külső-ROM/RAM
#6	$fd00-$ff3f	64768-65343		I/O terület & TED-regiszter	
#7	$ff49-$ffff	65344-65535		KERNAL vektorok	

Az egyes zónák tartalma:

zóna#1 16K RAM, benne a 6510 I/O-kapú regiszterei ($0000-$0001). Processzor-stack ($0100-$01ff), továbbá 2K képernyő-memória, 1K operációs-rendszer terület, BASIC terület és (grafikus módban) 8K bit-térkép.

zóna#2: üres, de átkapcsolható külső RAM-ra, megfelelő tárbővítő modullal (Pl.SZINTETORG Gmk. 64k-s tárbővítője, forgalmazza a FOTOELEKTRONIK-NOVOTRADE GT.). A tárbővítő a BASIC munkaterületét max. 60671 byte-ra növelheti, melynek kezelését a már említett RAM-olvasó rutinok problémamentesen végzik (a TED $ff3e, $ff3f regisztereivel a ROM-ot ki-be kapcsolják).

zóna#3 20K BASIC-ROM, de kikapcsolható, az alatta levő esetleges RAM-bővítés olvasható.

zóna#4 2k karakter-ROM

zóna#5 10k KERNAL-ROM, de az alatta levő RAM a speciális rutinokkal olvasható.

A ROM-ok kezelésében kiemelkedő szerepe van még a $fdd0-című TED-regiszternek, amelybe a megfelelő bitkombinációt (ROM-térkép) beírva a kívánt ROM terület címezhető.
Lehetséges bitkombinációk:

```
0-1 bitek: 0 - BASIC
           1 - -
           2 - C1 low
           3 - C2 low

2-3 bitek: 0 - KERNAL
           1 - -
           2 - C1 high
           3 - C2 high
```

6. KÜLSŐ CSATLAKOZÓK:

6.1. CARTRIDGE-csatlakozó (50 pólus, anya)

1 - Gnd	13 - BA	A - Gnd	P - A8
2 - +5V	14 - D7	B - C1-low	R - A7
3 - +5V	15 - D6	C - -RESET	S - A6
4 - IRQ	16 - D5	D - -RAS	T - A5
5 - R/W	17 - D4	E - O0	U - A4
6 - C1-high	18 - D3	F - A15	V - A3
7 - C2-low	19 - D2	H - A14	W - A2
8 - C2-high	20 - D1	J - A13	X - A1
9 - -CS1	21 - D0	K - A12	Y - A0
10 - -CS0	22 - AEC	L - A11	Z - NC
11 - -CAS	23 - EXT-AUDIO	M - A10	AA - NC
12 - MUX	24 - O2	N - A9	BB - NC
	25 - Gnd		CC - Gnd

6.2. AUDIO/VIDEO (DIN, 8 pólus, anya)

1 - LUM
2 - Gnd
3 - AUI.OUT
4 - COMP.VIDEO
5 - AUD.IN
6 - COLOR
7 - NC
8 - +5V

6.3. Kazettás szalagegység (MIN DIN, 8 pólus, anya)

1 - CRT (Gnd)
2 - +5V
3 - motor
4 - read
5 - write
6 - sense
7 - Gnd

6.4. IEC soros busz (DIN, 6 pólus)

1 - SErvice reQuest
2 - Gnd
3 - ATN
4 - CLK
5 - DATA
6 - RESET

6.5. Botkormány (MIN DIN, 8 pólus, anya)

1 - K0
2 - K1
3 - K2
4 - K3
5 - +5V
6 - K6 (Joy1) / K7 (Joy2)
7 - Gnd
8 - D2 (Joy1) / D1 (Joy2)

A TEDMON monitor gépi kódú programok készítését, kipróbálását segítő eszköz.
Meghívása:

BASIC-ből: MONITOR utasítással
Gépi-kódból: JMP $fff52

A TEDMON parancsai egy egy-karakteres utasításkódból és egy vagy több paraméterből állnak, amelyeket egy-egy szóköz választ el egymástól. A paraméterek 4 dígites hexadecimális címek ($-jel nélkül), 2 dígites hexadecimális byte(sorozat)-ok vagy karaktersorozatok lehetnek (kivétel az A(ssemble) parancs, lsd. ott).

A parancsok azonnal végrehajtódnak. Választható funkciók

funkció	hatás	szintaxis
Assemble	1 gépi kódú utasítás lefordítása és beírása a memóriába. A paraméterek formája: cím: betöltési-cím(hexa): xxxx opkód: 3 karakteres mnemonik operandus: konstans(hexa) #$xx cím(0-lapon): $xx cím(általában): $xxxx Megj: a relatív ugrásoknál is az abszolút címet kell megadni	A cím opkód operandus . cím opkód operandus
Disassemble	A kcím kezdetű, vcím-kcím méretű memóriaterület visszafejtése utasításokká	D [kcím [vcím]]
Listázás/módosítás		
Memory	Memória tartalmának listázása és módosítása karakteres és hexadecimális formában. A kijelzett tartalmat billentyűzetről felülírva, RETURN hatására a memória tartalma módosul.	M [kcím [vcím]] >cím [xx ...]
Register	A processzor regiszterei (PC, SR, A, X, Y, SP) tartalmának kijelzése, illetve a tartalmuk módosítása felülírással	R ;pc sr ar xr yr sp
összehasonlítás		
Compare	A kcím1 kezdetű, vcím1-kcím1 méretű terület összehasonlítása a kcím2-vel kezdődő, azonos nagyságú területtel.	C kcím1 vcím1 kcím2
Feltöltés		
Fill	A kcím kezdetű, vcím-kcím byte-os memóriaterület feltoltése a megadott hexadecimális értékkel	F kcím vcím byte
Keresés		
Hunt	Bytesorozat /szöveg keresése a kcím kezdetű, vcím-1 végcímű memóriaterületen	H kcím vcím [xx ...] H kcím vcím 'szöveg
átmásolás		
Transfer	kcím1-től vcím1-1-ig terjedő terulet átmásolása a kcím2-től kezdődő teruletre	T kcím1 vcím1 kcím2
Save	Memóriatartalom kimentése háttértárolóra kcím-től vcím-1-ig	S"filenév",08,kcím,vcím S"filenév",01,kcím,vcím
Load	Program betöltése háttértárolóról a kimentés helyére	L"filenév",08 L"filenév",01
Verify	Memóriatartalom és háttértárra kiírt másolatának összevetése (lsd. BASIC VERIFY parancsát)	V"filenév",08 V"filenév",01
Rutinhívás		
Go	Vezérlésátadás a megadott címen vagy PC aktuális értékénél kezdődő gépi kódú programrészre	G [cím]
Exit	Visszatérés a BASIC-hez	X

A. SZÍNKÓDOK

Szín	Billentyű	Szín-szám (COLOR-ban)	Szín - kód (belső ábrázolás)	Idézőjel módban
fekete	ctrl+1	1	0	
fehér	ctrl+2	2	1	
piros	ctrl+3	3	2	
cián	ctrl+4	4	3	
bíbor	ctrl+5	5	4	
zöld	ctrl+6	6	5	
kék	ctrl+7	7	6	
sárga	ctrl+8	8	7	
narancs	C= + 1	9	8	
barna	C= + 2	10	9	
sárgászöld	C= + 3	11	10	
rózsaszín	C= + 4	12	11	
kékeszöld	C= + 5	13	12	
világoskék	C= + 6	14	13	
sötétkék	C= + 7	15	14	
vil.zöld	C= + 8	16	15	

B. SZÍNKOMBINÁCIÓK

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	.	x	.	x	x	m	.	x	x	.	x	x	m	x	.	x
1	x	.	x	.	x	x	x	.	m	x	m	m	x	x	x	.
2	.	x	.	.	m	.	.	x	x	.	.	x	.	.	.	m
3	x	.	.	.	.	m	x	.	.	.	.	.	m	m	x	.
4	x	m	.	.	.	.	.	.	.	.	.	.	.	.	m	.
5	x	m	.	m	.	.	.	.	.	.	x	.	.	.	.	x
6	m	x	.	x	.	.	.	.	.	.	m	.	.	x	.	m
7	x	.	x	.	.	.	m	.	m	x	.	.	.	.	.	.
8	m	x	x	.	.	.	.	m	.	x	.	.	.	.	.	.
9	m	x	.	.	.	.	.	x	x	.	x	x	.	.	.	m
10	x	.	.	.	.	x	m	.	.	x	.	.	.	.	m	m
11	x	m	x	.	.	.	.	m	.	x	.	.	.	.	m	m
12	m	x	.	m	.	.	.	.	.	.	.	.	.	.	m	m
13	x	x	.	m	.	.	.	.	.	.	.	.	.	.	x	m
14	.	x	.	x	.	.	m	.	.	.	m	m	m	x	.	m
15	x	.	.	.	.	x	m	.	.	.	m	m	m	m	m	.

x - kitűnő m - közepes . - rossz

C. A LENYOMOTT BILLENTYŰ KÓDJA (LOC. 198 / $C6 TARTALMA):

	0	1	2	3	4	5	6	7	8	9
0	del	rtn		£	help	f1	f2	f3	@	3	W
1		A	4	Z	S	E		5	R	D	6
2		C	F	T	X	7	Y	G	8	B	H
3		U	V	9	I	J	0	M	K	O	N
4	cud		P	L	cup	.	:	-	,	cul	*
5		;	cur	esc	=	+	/	1	hom		2
6	spc			Q		none					

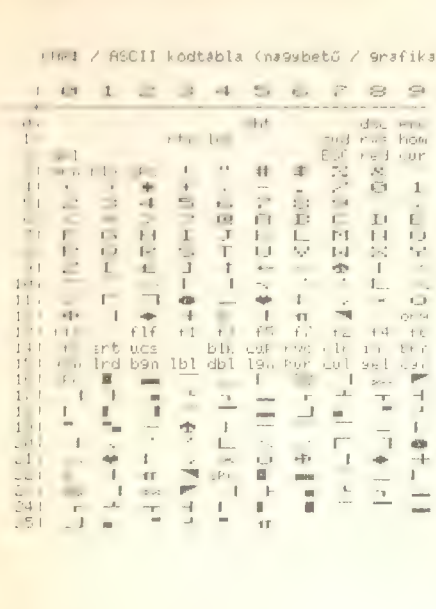
cup: kurzor fel
cur kurzor jobb
cud kurzor le
cul kurzor balra
spc szóköz

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	del	rtn		£	help	f1	f2	f3	@	3	W	A	4	Z	S	E	
1		5	R	D	6	C	F	T	X	7	Y	G	8	B	H	U	V
2		9	I	J	0	M	K	O	N	cud	P	L	cup	.		-	,
3	cul	*		;	cur	esc	=	+	/	1	hom		2	spc		0	
4	none																

D. ASCII KARAKTEREK(decimális táblázat):

KAR# / ASCII kódtábla (nagybetű / grafika)

	0	1	2	3	4	5	6	7	8	9
0									dsc	enc
1				rtn	lcs			cud	rvs	hom
2	del							ESC	red	cur
3	grn	blu	spc	!	"	#	$	%	&	
4	'	'	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9		.
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	£	]	↑	←		♠		
10										
11										
12										org
13	flo		flf	f1	f3	f5	f7	f2	f4	f6
14	f8	srt	ucs		blk	cup	rvo	clr	ins	brn
15	ygn	lrd	bgn	lbl	dbl	lgn	pur	cul	yel	cyn
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										



Jelmagyarázat

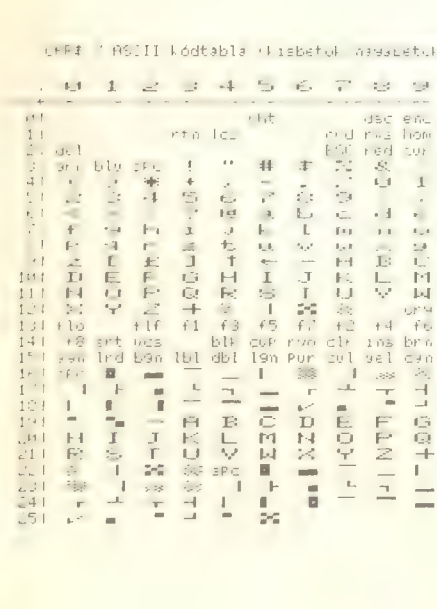
Standard
ASCII vezérlő karakterek

NUL	Null
SOH	Start Of Heading
STX	Start Of Text
ETX	End Of Text
EOT	End Of Transmiss.
ENQ	Enquiry
ACK	Acknowledge
BEL	Bell
BS	Backspace
HT	Horizontal Tab
LF	Line Feed
VT	Vertical Tab
FF	Form Feed
CR	Carriage Return
SO	Shift Out
SI	Shift In
DLE	Data Link Escape
DC	Device Control
NAK	Neg.Acknowledge
SYN	Synchronous Idle
ETB	End Of Transm.Bl.
CAN	Cancel
EM	End Of Medium
SUB	Substitute
ESC	Escape
FS	File Separator
GS	Group Separator
RS	Record Separator
US	Unit Separator
spc	space/szóköz
bsl	back-slash-jel
bkk	bal kapocs
jkk	jobb-kapocs
haz	háztető-jel
DEL	Delete

KAR# / ASCII kódtábla (kisbetők nagybetők)

	0	1	2	3	4	5	6	7	8	9
0									dsc	enc
1				rtn	lcs			cud	rvs	hom
2	del							ESC	red	cur
3	grn	blu	spc	!	"	#	$	%	&	
4	'	'	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9		.
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	£	]	↑	←		♠		
10	D	E	F	G	H	I	J	K	L	M
11	N	O	P	Q	R	S	T	U	V	W
12	X	Y	Z	+						org
13	flo		flf	f1	f3	f5	f7	f2	f4	f6
14	f8	srt	ucs		blk	cup	rvo	clr	ins	brn
15	ygn	lrd	bgn	lbl	dbl	lgn	pur	cul	yel	cyn
16										
17										
18										
19				A	B	C	D	E	F	G
20	H	I	J	K	L	M	N	O	P	Q
21	R	S	T	U	V	W	X	Y	Z	+
22					spc					
23										
24										
25										

COMMODORE
Vezérlő karakterek



wht	white/fehér
bgn	blu-grn/kékeszöld
blk	black/fekete
blu	blue/kék
brn	brown
clr	CLeaR
cud	cursor down
cul	cursor left/bal
cup	cursor up/fel
cur	cursor right/jobb
cyn	cyan/kékeszöld
dbl	dark-blue/sötétkék
del	DELete
dsc	SHIFT C= letiltása
enc	SHIFT C= megenged.
esc	escape
flo	flash on
flf	flash off
grn	green/zöld
hom	HOME
ins	INSerT
lbl	light blue/v.kék
lcs	lowercase/kisbetű
lgn	light green/v.zöld
lrd	light red/v.piros
org	orange/narancs
pur	purple/bíbor
red	red/piros
rtn	RETURN
rvo	reverse off
rvs	reverse on
spc	space/szóköz
srt	SHIFT RETURN
ucs	uppercase/nagybetű
yel	yellow/sárga
ygn	yel-grn/sárgászöld

ASCII KARAKTEREK(hexadecimális táblázat)

CHR$ / ASCII kódtábla (nagybetu / grafika)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0						wht			dsc	enc				rtn	lcs	
1		cud	rvs	hom	del							ESC	red	cur	grn	blu
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	£	]	↑	←
6	─	♠	┃	─	┃	─	┼	┃	╲	╱	◥	┘	└	┼	╱	╲
7	┐	●	▁	◆	▕	▔	╳	○	♣	▕	◆	┼	▒	▕	┼	▗
8		org	flo		flf	f1	f3	f5	f7	f2	f4	f6	f8	srt	ucs	
9	blk	cuP	rvo	clr	ins	brn	y9n	lrd	b9n	lbl	dbl	l9n	Pur	cul	yel	cyn
A	SPC	▘	▀	┴	▁	▌	▒	▁	▀	◤	▗	▝	▪	▟	▝	▛
B	▗	┴	┬	┤	▏	▙	▐	▔	▀	▙	▟	▪	▝	▟	▛	▙
C	─	♠	┃	─	▔	▁	╳	○	♣	◢	◆	┼	▒	▗	▛	◢
D	┐	●	▁	◆	▕	▔	╳	○	♣	◢	◆	┼	▒	▗	▛	▗
E	SPC	▘	▀	┴	▁	▌	▒	▁	▀	◤	▗	▝	▪	▟	▝	▛
F	▗	┴	┬	┤	▏	▙	▐	▔	▀	▙	▟	▪	▝	▟	▛	▙

CHR$ / ASCII kódtábla (kisbetük/nagybetük)

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0						wht			dsc	enc				rtn	lcs	
1		cud	rvs	hom	del							ESC	red	cur	grn	blu
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	P	q	r	s	t	u	v	w	x	y	z	[	£	]	↑	←
6	─	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
7	P	Q	R	S	T	U	V	W	X	Y	Z	+	▒	▕	▚	▒
8		org	flo		flf	f1	f3	f5	f7	f2	f4	f6	f8	srt	ucs	
9	blk	cuP	rvo	clr	ins	brn	y9n	lrd	b9n	lbl	dbl	l9n	Pur	cul	yel	cyn
a	SPC	▘	▀	┴	▁	▌	▒	▁	▀	▒	▗	▝	▪	▟	▝	▛
b	▗	┴	┬	┤	▏	▙	▐	▔	▀	▙	▟	▪	▝	▟	▛	▙
c	─	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
d	P	Q	R	S	T	U	V	W	X	Y	Z	+	▒	▕	▚	▒
e	SPC	▘	▀	┴	▁	▌	▒	▁	▀	▒	▗	▝	▪	▟	▝	▛
f	▗	┴	┬	┤	▏	▙	▐	▔	▀	▙	▟	▪	▝	▟	▚	▒

E. HEXADECIMALIS-DECIMÁLIS KONVERZIÓ:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

F. KEPERNYO-KODOK

(128–255 is a 0-127 karakterek inverzei jelennek meg

KEPERNYO-KODOK KARAKTERKESZLET 1

	0	1	2	3	4	5	6	7	8	9
0	@	A	B	C	D	E	F	G	H	I
1	J	K	L	M	N	O	P	Q	R	S
2	T	U	V	W	X	Y	Z	[	£	]
3	↑	←	SPC	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	─	♠	┃	─	─	─
7	─	┃	┃	╲	╲	╱	╲	╲	╱	┌
8	┐	●	─	♥	┃	╱	╳	○	♣	┘
9	♦	+	▒	┃	π	◥	SPC	▮	▬	▬
10	─	┃	▒	┃	▒	◤	┃	┣	▬	╰
11	┐	─	┌	┴	┬	┤	┃	▌	▐	─
12	─	▬	┘	▄	▀	┘	▀	▄		

KEPERNYO-KODOK KARAKTERKESZLET 2

	0	1	2	3	4	5	6	7	8	9
0	@	a	b	c	d	e	f	g	h	i
1	j	k	l	m	n	o	p	q	r	s
2	t	u	v	w	x	y	z	[	£	]
3	↑	←	SPC	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	─	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	+	▒	┃	▒	▒	SPC	▮	▬	─
10	─	┃	▒	┃	▒	▒	┃	┣	▬	╰
11	┐	─	┌	┴	┬	┤	┃	▌	▐	─
12	─	▬	╱	▄	▀	┘	▀	▄		

KEPERNYOKODOK KARAKTERKESZLET 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	£	]	↑	←
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	─	♠	┃	─	─	─	─	┃	┃	╲	╲	╱	╲	╲	╱	┌
5	┐	●	─	♥	┃	╱	╳	○	♣	┘	♦	+	▒	┃	π	◥
6	SPC	▮	▬	─	─	┃	▒	┃	▒	▀	┘	┣	▀	┘	┐	▬
7	┌	┴	┬	┤	┃	▌	▐	─	▬	▬	┘	▄	▀	┘	▀	▄

KEPERNYOKODOK KARAKTERKESZLET 2

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	@	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
1	p	q	r	s	t	u	v	w	x	y	z	[	£	]	↑	←
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	─	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	+	▒	┃	▒	▒
6	SPC	▮	▬	─	─	┃	▒	┃	▒	▒	┘	┣	▀	┘	┐	▬
7	┌	┴	┬	┤	┃	▌	▐	─	▬	▬	╱	▄	▀	┘	▀	▄

G. BASIC KULCSSSZAVAK, RÖVIDÍTESÜK ÉS TOKENJEIK (alfabetikus sorrend):

kulcsszó	röv.	típ	token	kulcsszó	röv.	típ	token
abs	aB	f#	182 $b6	new		p	162 $a2
and	aN	o	175 af	next	nE	u	130 82
asc	aS	f#	198 c6	not	nO	o	168 a8
atn	aT	f#	193 c1	on		u	145 91
auto	aU	p	220 dc	open	oP	u	159 9f
backup	bA	p	246 f6	or		o	176 b0
box	bO	u	225 e1	Paint	pA	u	223 df
char	chA	u	224 e0	Peek	pE	f#	194 c2
chr$	cH	f$	199 c7	Poke	pO	u	151 97
circle	cI	u	226 e2				
close	clO	u	160 a0	Pos		f#	185 b9
clr	cL	u	156 9c	Print	?	u	153 99
cmd	cM	u	157 9d	Print#	pR	u	152 98
collect	colL	p	243 f3	Pudef	pU	u	221 dd
color	coL	u	231 e7	rclr	rC	f#	205 cd
cont	cO	p	154 9a	rdot	rD	f#	208 d0
copy	coP	p	244 f4	read	rE	u	135 87
cos		f#	190 be	rem		u	143 8f
data	dA	u	131 83	rename	reN	p	245 f5
dec	-	f#	209 d1	renumber	renU	p	248 f8
def	dE	u	150 96	restore	reS	u	140 8c
delete	deL	p	247 f7	resume	resU	u	214 d6
dim	dI	u	134 86	return	reT	u	142 8e
directory	diR	p	238 ee	rgr	rG	f#	204 cc
dload	dL	p	240 f0	right$	rI	f$	201 c9
do		u	235 eb				
draw	dR	u	229 e5	rlum	rL	f#	206 ce
dsave	dS	p	239 ef	rnd	rN	f#	187 cb
ds		v					
el		v					
end	eN	u	128 80	run	rU	p	138 8a
er		v					
err$	eR	f$	211 d3	save	sA	p	148 94
exit	exI	u	237 ed				
exp	eX	f#	189 bd	scale	scA	u	233 e9
fn		f#	165 a5	scnclr	sC	u	232 e8
for	fO	u	129 81	scratch	scR	p	242 f2
fre	fR	f#	167 a7	sgn	sG	f#	180 b4
get	gE	u	161 a1	sin	sI	f#	191 bf
getkey	getkE	u	161+	sound	sO	u	218 da
get#	gE#	u	161+	spc(	sP	edit	166 a6
go		u	203 cb				
gosub	goS	u	141 8d	sqr	sQ	f#	186 ba
goto	gO	u	137 89	sshape	sS	u	228 e4
graphic	gR	u	222 de	status	st	v	
gshape	gS	u	227 e3	step	stE	m	169 a9
header	heA	p	241 f1	stop	sT	u	144 90
hex$	hE	f$	210 d2	str$	stR	f$	196 c4
if		u	139 8b	sys	sY	u	158 9e
input		u	133 85	tab(	tA	edit	163 a3
input#	iN	u	132 84	tan		f#	192 c0
instr	inS	f#	212 d4	then	tH	u	167 a7
int		f#	181 b5	time	ti	v	
joy	jO	f#	207 cf	time$	ti$	v$	
key	kE	u	249 f9	to		u	164 a4
left$	leF	f$	200 c8	trap	tR	u	215 d7
len		f#	195 c3	troff	troF	u	217 d9
let	lE	u	136 88	tron	trO	u	216 d8
				until	uN	u	252 fc
list	lI	p	155 9b	using	usI	edit	251 fb
load	lO	p	147 93	usr	uS	f#	183 b7
locate	loC	u	230 e6	val	vA	f#	197 c5
log		f#	188 bc	verify	vE	p	149 95
loop	loO	u	236 ec	vol	vO	u	219 db
mid$	mI	f$	202 ca	wait	wA	u	146 92
monitor	mO	u	250 fa	while	wH	u	253 fd
+		o	170 aa	>		o	177 b1
-		o	171 ab	=		o	178 b2
*		o	172 ac	<		o	179 b3
/		o	173 ad				

u - BASIC utasítás
p - Parancs
f# - numerikus függvény
f$ - string-függvény
v - foglalt BASIC változónév
v$ - stringváltozó
o - műveleti jel
edit - output-szerkesztő függvény

BASIC kulcsszavak és tokenjeik (numerikus sorrend):

token	kulcsszó	token	kulcsszó	token	kulcsszó
128 $80	end	171 $ab	-	214 $d6	resume
129 81	for	172 ac	*	215 d7	trap
130 82	next	173 ad	/	216 d8	tron
131 83	data	174 ae	↑	217 d9	troff
132 84	input#	175 af	and	218 da	sound
133 85	input	176 b0	or	219 db	vol
134 86	dim	177 b1	>	220 dc	auto
135 87	read	178 b2	=	221 dd	pudef
136 88	let	179 b3	<	222 de	graphic
137 89	goto	180 b4	sgn	223 df	paint
138 8a	run	181 b5	int	224 e0	char
139 8b	if	182 b6	abs	225 e1	box
140 8c	restore	183 b7	usr	226 e2	circle
141 8d	gosub	184 b8	fre	227 e3	gshape
142 8e	return	185 b9	pos	228 e4	sshape
143 8f	rem	186 ba	sqr	229 e5	draw
144 90	stop	187 bb	rnd	230 e6	locate
145 91	on	188 bc	log	231 e7	color
146 92	wait	189 bd	exp	232 e8	scnclr
147 93	load	190 be	cos	233 e9	scale
148 94	save	191 bf	sin	234 ea	help
149 95	verify	192 c0	tan	235 eb	do
150 96	def	193 c1	atn	236 ec	loop
151 97	poke	194 c2	peek	237 ed	exit
152 98	print#	195 c3	len	238 ee	directory
153 99	print	196 c4	str$	239 ef	dsave
154 9a	cont	197 c5	val	240 f0	dload
155 9b	list	198 c6	asc	241 f1	header
156 9c	clr	199 c7	chr$	242 f2	scratch
157 9d	cmd	200 c8	left$	243 f3	collect
158 9e	sys	201 c9	right$	244 f4	copy
159 9f	open	202 ca	mid$	245 f5	rename
160 a0	close	203 cb	go	246 f6	backup
161 a1	get/get#	204 cc	rgr	247 f7	delete
162 a2	new	205 cd	rclr	248 f8	renumber
163 a3	tab(	206 ce	rlum	249 f9	key
164 a4	to	207 cf	joy	250 fa	monitor
165 a5	fn	208 d0	rdot	251 fb	using
166 a6	spc(	209 d1	dec	252 fc	until
167 a7	then	210 d2	hex$	253 fd	while
168 a8	not	211 d3	err$	254 fe	usrtoken
169 a9	step	212 d4	instr		
170 aa	+	213 d5	else		

H. A MÁGNESLEMEZ MINT ADATHORDOZÓ

A VC1541 un. soft-szektoros, egyoldalas, szimpla sűrűségű lemezeket használ (soft-sectored, single density, single side). A mágneslemez adattároló felülete fizikailag 35 koncentrikus SÁVra oszlik, amelyek kívülről befelé az 1-35 sorszámokat viselik. A sávok 256-byte-os SZEKTORokra, más néven BLOKKokra vannak felosztva, az alábbi elrendezésben:

```
 1-17 sáv: 21 szektor
18-24 sáv: 19 szektor
25-30 sáv: 18 szektor
31-35 sav: 17 szektor
```

összesen 683 blokk, ebből 664 blokk áll a felhasználó rendelkezésére.

H.1. A katalógus (directory):

A 18. sávot foglalja el. Lényegében szekvenciális file, amely a "$" filenévvel érhető el. Három fő szerkezeti elemből áll:

BAM (Block Availability MaP) vagyis blokk-foglaltsági térkép
Katalógus-header
Katalógus-bejegyzések

A BAM szerkezete (18.sáv 0.szektor)

byte	tartalom	definíció
0-1	$12,$01	első katalógus-blokk címe (sáv,szektor)
2	$41	ASCII"A" 4040 formátum jelzésére
3	$00	Jövőbeni használatra lefoglalva
4-143	------>	1-35 sávok foglaltsági bit-térképe: 1 - üres blokk 0 - foglalt blokk

A katalógus-HEADER szerkezete (18.sáv 0.szektor):

byte	tartalom	definíció
144-161	------>	diszk-név, Jobbról $a0-ákkal feltöltve
162-163	------>	diszk-ID
164	$A0	
165-166	$32,$41	"2A" DOS verzió és formátum típus
166-167	$A0,$A0	
171-255	------>	nem használt

A katalógus-BLOKKOK szerkezete (18.sáv 1-18 szektor):

byte	tartalom
0-1	következő katalógus-blokk címe(sáv,szektor)
2-31	1. bejegyzés
34-63	2. "
66-95	3. "
98-127	4. "
130-159	5. "
162-191	6. "
194-223	7. "
226-255	8. "

A katalógus-BEJEGYZÉSEK szerkezete:

byte	tartalom
0	file-típus: - ha "rendesen" le lett zárva, $80-nal OR-olva - $40-nel OR-olva védett állomány, letörölhetetlen! típusok: 0 - törölt file (DEL) 1 - szekvenciális (SEQ) 2 - program (PRG) 3 - user (USR) 4 - relatív (REL)
1-2	az első adatblokk sáv és szektorcíme
3-18	file-név, Jobbról $A0-val feltöltve
19-20	csak REL file-oknál: első "side"-szektor címe
21	csak REL file-oknál: rekord mérete
22-25	használatlan
26-27	a helyettesítő (replacement) file sáv és szektorcíme, ha OPEN@ van folyamatban
28-29	a file hossza blokkokban (low-byte,hi-byte)

N.1. File-ok

SZEKVENCIÁLIS FILE szerkezete

byte	tartalom
0-1	következő adatblokk címe (sáv,szektor) (az utolsó adatblokkban: byte 0: 0 byte 1: utolsó használt byte offszetje
2-255	254 byte adat

PROGRAM FILE szerkezete:

byte	tartalom
0-1	következő program-blokk címe(sáv,szektor)
2-255	254 byte program CBM memória formátumban, kódolt kulcsszavakkal. File-vége jel: $000000 (az első blokkban: byte 2-3: a program betöltési címe kimentéskor)

RANDOM blokk szerkezete:

byte	tartalom
0	blokk kurrens hossza(ha B-W-vel lett kiírva)
1-255	adat

RELATÍV FILE szerkezete

KATALÓGUS-BLOKK(side-sector): file-onként max. 6 db.

byte	tartalom
0-1	következő katalógus-blokk címe(sáv,szektor)
2	katalógus-blokk sorszáma (0-5)
3	rekord-hossz
4-5	0-ás katalógusrekord címe(sáv,szektor)
6-7	1-es katalógusrekord címe(sáv,szektor)
8-9	2-es katalógusrekord címe(sáv,szektor)
10-11	3-as katalógusrekord címe(sáv,szektor)
12-13	4-es katalógusrekord címe(sáv,szektor)
14-15	5-ös katalógusrekord címe(sáv,szektor)
16-255	120 db. adatblokk címe(sáv,szektor)

ADATBLOKK

byte	tartalom
0-1	következő adat-blokk címe(sáv,szektor)
2-255	254 byte adat. Az üres rekord $FF kezdőbyte után csupa 0-bCl áll. A részben töltött rekordok üres része szintén 0-val van feltöltve.

I. A LEMEZEGYSÉG HIBASTATUSZAI

0 - nincs hiba
1 - file-ok törölve
20 - hiányzik a blokk-header (lemezhiba v. egységhiba)
21 - szinkron-karakter hiányzik
22 - adatblokkot nem találta (nem jól van felírva)
23 - kontrolszumma hiba (lemez v.földelési hiba)
24 - byte-dekódolási hiba (lemez v.földelési hiba)
25 - hibát talált írás-ellenőrzésnél
26 - írás-próbálkozás írásvédett lemezre
27 - kontrolszumma-hiba blokkfejben (lemez v.földelési hiba)
28 - adat folytatódik a következő blokkban (lemez v. hw hiba
29 - lemez-azonosító (ID) ellentmondás (lemezhiba)
30 - szintaktikus hiba
31 - érvénytelen utasítás (felesleges szóköz a Parancs elején)
32 - hosszú sor (Parancs hosszabb mint 58 karakter)
33 - érvénytelen file-név
34 - hiányzik a file-név
39 - Parancs-file hiányzik
50 - nincs több rekord (file-vége volt)
51 - túlcsordulás a rekordban(CHR$(13) is része a rekordnak)
52 - file túl nagy (relatív file-nál)
60 - file írásra van megnyitva
61 - file nincs megnyitva
62 - nincs a file a lemezen
63 - a file már létezik
64 - file-típus ellentmondás
65 - a blokk foglalt (B-A utasításnál)
66 - illegális sáv vagy szektor
67 - illegális rendszer-sáv vagy szektor
70 - nincs csatorna (max. 5 soros v. 6 random file lehet nyitva)
71 - katalógus-hiba (az egység újra INITIALIZE-andó)
72 - lemez vagy katalógus tele van
73 - lemez-formátum hiba (másik DOS-verzió)
74 - nincs lemez az egységben

J. PROGRAMOZÁSI PÉLDÁK

J.1 Várakozás billentyű leütésére a WAIT utasítás felhasználásával

```
10 PRINT "USS LE EGY BILLENTYUT"
20 WAIT 198,191:  REM HA NINCS, 198 TARTALMA 64 (BIT 7 ON)
30 PRINT "KOSZONOM"
```

J.2. Példák a billentyűPuffer használatára:

onmódosító Pro9ram

```
10 PRINT"usd be X tetszole9es fu99venyet"
20 INPUT X$
30 POKE239,3:POKE1319,19:POKE1320,13:POKE1321,13
40 PRINTCHR$(147)"100DEFFNA(X)="X$":RETURN"
50 PRINT"GOTO 60":SYS 34578
55 REM 34578 a BASIC utasítás-olvasó rutinja
60 GOSUB 100
70 INPUT "kerem X erteket";X:PRINT"FN(X)="FNA(X):GOTO70
```

Basic-overlay

```
145 PRINTCHR$(147)"NEW":PRINTCHR$(17)CHR$(17)"LOAD"CHR$(34)
    "overlay"CHR$(34)",8"
170 POKE1319,19 POKE1320,13 POKE1321,13 POKE1322,19
174 POKE1323,82:POKE1324,117:POKE1325,32:POKE1326,13
175 POKE239,8:END
```

J.3. BASIC-96Pi-kód interface (Pszeudore9iszterek használata)

Példa: GépI-kódú rutin a billentyű-mátrix felhasználására, a leütött billentyű letaPo9atására (lehetővé teszi e9yszerre lenyomott több billentyű azonosítását is!):

```
*=$033d
sei                 interruPtot letiltjuk
                    A már be van töltve SAREG-ből (SYS)
sta $fd30           maszkot beírjuk a "row-select re9.be
sta $ff08           írunk bele, ho9y utána olvashassuk
lda $ff08           kiolvassuk a "column read" fla9-eket
cli                 interruPtot felszabadítjuk
                    A-t majd SYS visszaírja SAREG-be
rts                 visszatérÜnk a hívás helyére
```

A rutint hívó BASIC-Pro9ram:

```
1 INPUT "SOR=";S
2 POKE 2034,255-2↑S:       REM MASZK KITAROLASA SAREG-BE
3 SYS 829:                 REM SZUBRUTIN MEGHíVASA
4 PRINT S,255-PEEK(2034): REM FLAG-BYTE KIOLVASASA SAREG-BöL
5 GOTO 2
```

J.4. A raszter-interrupt felhasználása

Az interrupt lekezelésével a képernyő akár több, különböző üzemmódú részre osztható.

Rövid példa:

```
        *=  $0033   rutin a szalag-pufferben

 INIT sei           inhibit interrupt
      lda #$40      interrupt vektor
      sta $0314     módosítása rutinunk
      lda #$03      címére
      sta $0315
      cli
      rts           kész
 INTR lda $ff09     raszter innterrupt?
      and #$02
      beq NEMR      ha nem, ugrunk
      sta $ff09     IRQ-REG-et töröljük
      lda $ff19     raszter HI-bit on?
      and #$01
      bne NEMHI     ha nem, ugrunk
      ldy #$52      Piros mező
      ldx #$50      80. rasztersorig
      cpx $ff1d     elértük
      bcs HI        nem még
NEMHI ldy #$56      kék szín
      ldx #$40      60-as sorig
HI    stx $ff0b     raszter reg
      sty $ff15     szín módosítás
NEMP  jsr $cfbf     kazetta
      jsr $cecd     hang kitart.
      lda $fb       save ROM-map
      pha
      lda #$00      0-ás map
      sta $fb
      php
      cli
      jsr $db11     billentyűzet
      plp
      pla           ROM-map restore
      sta $fb
      jmp $fcbe     IRQ folytatása
```

A rutin aktiválása BASIC-ból

```
 SYS  819
```

J.5. BASIC szöveg feldolgozása gépi-kódból

PELDA#1: Készüljön szubrutin a képernyő-memória háttérszínének beállítására. A rutin hívási szintaxisa legyen:

```
            SYS belépési-Pont,kezdőcím,színkód
#=$nnnn
firstP=$14          Algoritmusa:

zback9 jsr chkoma       FOR I=BASE TO BASE+999
       jsr getcor          POKE I,SZINKOD
       lda #0           NEXT
       sta count
       sta count+1
       ldy #0
frissa txa                  újra betöltjük a színkódot
store  sta (firstP),y       beírjuk
       inc count            növeljük a számláló lo-byte-ját
       bne next             túlcsordult?
       inc count+1          igen, növeljük hi-byte-ot is
next   lda #>1000           elértük a felső határt?
       cmp count+1
       bne ujra             nem
       lda #<1000
       cmp count
       bne ujra             nem
       rts                  igen, visszatérünk
ujra   inc firstP           nem, növeljük az indirekt címet is
       bne frissa
       inc firstP+1
       jmp frissa
count  .byte $0,$0
```

A fenti rutin hívása BASIC-ből (Például):

```
10  zback9=DEC($nnnn) : kcim=4*256
20  SYS zback9,kcim,1
```

PÉLDA#2: Két paraméterpár átvétele (x1,y1,x2,y2)
Pl. egyenes rajzolásához:

```
zline  =*
       jsr chkoma
       jsr getcor       első paraméterpár X-ben és firstP-ben
       stx y1
       lda firstP
       sta x1
       lda firstP+1
       sta x1+1
       jsr chkoma       pozicionálás vesszőig
       jsr getcor       x2,y2 beolvasása
       stx y2
       lda firstP
       sta x2
       lda firstP+1
       sta x2+1
stb.
```

Hívása BASIC-ből:

```
10   LINE=DEC($mmmm)
450  SYS LINE,x1,y1,x1*2,y1+4*b
```

PÉLDA#3: x,y koordináta és függvényérték átvétele:

```
zuser  =*
       jsr chkoma
       jsr getcor     első paraméterpár x-ben és firstP-ben
       stx y          lásd zline
       lda firstP
       sta x1
       lda firstP+1
       sta x1+1
       jsr chkoma     pozicionálás
       jsr frnum      lebegőpontos paraméter FAC=ban
stb.
```

Hívása BASIC-ből: (Például)

```
670  SYS zuser,x,y,sin(x)
```

PÉLDA#4: számítsuk ki a következő kifejezés értékét:

cz=az↑2+bz

```
.
.
.
lda #$az-cím-lo  (cím alsó byte-ja)
ldy #$az-cím-hi  (cím felső byte-ja)
jsr ldfac1     FAC=az
lda #$az-cím-lo
ldy #$az-cím-hi
jsr mpymf1     FAC=az*FAC
lda #$bz-cím-lo
ldy #$bz-cím-hi
jsr addmf1     FAC=bz+FAC
ldx #$cz-cím-lo
ldy #$cz-cím-hi
jsr stfac1     cz=az*az+bz
.
.
.
```

K. A ROM ÉS A RAM TÉRKÉPE

*** BASIC INTERPRETER **********

```
8000 BASIC hideg indítás
8003 BASIC melegindítás
8007 CBM - ROM azonosító
802e BASIC-RAM incializálás
80cc Bekapcsolási üzenet
8105 BASIC vektor táblázat
8117 BASIC vektorok RAM-ba másolása
818e BASIC kulcsszavak
8183 BASIC utasítások címei (-1)
8415 BASIC függvények címei
8453 Műveleti hierarchia és címek(-1)
8471 BASIC hibaüzenetek
8653 BASIC hibaüzenet címkiszámítás
866f READY kiírása
867e BASIC újraindítás
8a03 token azonosítása
8a79 BASIC NEW
8a98 BASIC CLR
8aff BASIC LIST
8bbc BASIC RUN
8bc4 RUN sorszámmal
8c6b MID$ az értékadás bal oldalán
8c74 Magában álló GO
8c82 User token értelmezés
8c9a BASIC RESTORE
8cc0 STOP billentyű figyelése
8cd8 BASIC STOP
8cda BASIC END
8d03 BASIC CONT
8d2c BASIC GOSUB
8d4d BASIC GOTO
8d83 BASIC RETURN
8db0 BASIC DATA
8de1 BASIC IF
8e0b BASIC REM, ELSE (nem végrehajtás esetén)
8e10 BASIC ELSE (végrehajtás esetén)
8e1b BASIC ON
8e7c BASIC LET
8fe0 BASIC PRINT#
8fe6 BASIC CMD
9000 BASIC PRINT
905f BASIC TAB(, SPC(
90b8 BASIC GET
90ee BASIC INPUT#
9108 BASIC INPUT
914f BASIC READ
9294 BASIC NEXT
9314 FRMNUM - num. kifejezés beolv.
932c FRMEVL - kifejezés kiértékelés
9414 Kifejezés következő eleme
9465 BASIC NOT
9477 BASIC FN
947e BASIC SGN
94ad BASIC változó kiértékelése
94e8 BASIC DS$ változó
94fa Diszk státusz olvasása
9501 Numerikus változó kiértékelése
9531 A TI$ betöltése
953f BASIC ST változó
954d BASIC DS változó
9577 BASIC ER, EL változó
9599 Függvény kiértékelése
95b4 Szövegfüggvény kiértékelése
95d9 Numerikus függvény kiértékelése
95f8 BASIC OR
95fb BASIC AND
9628 BASIC relációk (<, >, =)
9640 Szöveg összehasonlítás
969e BASIC DIM
96a5 változó keresése
9871 FAC mint INT az (A/Y)-ba
9879 REAL/INT
989b Tömbelem beolvasása
993a Tömbváltozó tárolása
99c2 Tömbelem keresése
99f1 Tömbelem címének kiszámítása
9a62 BASIC FRE
9a7d BASIC POS
9a86 Parancsmód teszt
9a92 Integer A/Y az akkuba
9a9d BASIC DEF
9acb FN szintaxis ellenőrzése
9b54 Szövegmutató kiszámítása
9b66 BASIC STR$
```

********** RAM **********

```
0000 I/O Port adatirány regiszt.
0001 I/O Port
0002 BASIC veremkezelés
0003-0006 konverzió munkaterület
0007-0008 munkaterület
0009 TAB-Pozíció
000a LOAD/VERIFY
000b DIM szám
000c DIM flag
000d szám/szöveg kapcsoló (ff:$)
000e egész/float kapcsoló (80:%)
000f idézőjel-mód flag
0011 READ/INPUT/GET
0013 INPUT-egység
0014-0015 munkaterület
     konverzió,Ptr
0016 szövegverem-mutató
0017-0021 szövegverem
0022-002a munkaterület
002b-002c BASIC tár kezdete
002d-002e változóterület kezdete
002f-0030 változóterület vége
0031-0032 tömbterület vége
0033-0034 szöveg terület kezdete
0035-0036 szöveg segédmutató
0037-0039 BASIC tár vége
0039-003a aktuális sor száma
003b-003c aktuális sor címe
003d-003e ideiglenes veremmutató
003f-0040 DATA aktuális sorszám
0041-0042 következő DATA-elem
0043-0044 INPUT-mutató
0045-0046 aktuális változó neve
0047-0048 aktuális változó címe
0049-004a aktuális vált. értéke
004b-004c Programmutató(ideigl)
004d összehasonlító operátor maszk
004e-004f FN-mutató
0050-0053 string-leíró
0054     JMP
0055-0056 függvény vektor
0057-005b FAC#3
005c-0060 FAC#4
0061-0065 FAC#1
0066 FAC előjel
0067 Polinom-kiértékelés száml.
0068 FAC kerekítő érték
0069-006d FAC#2 (ARG)
006e ARG előjel
006f FAC/ARG előjel-hasonlítás
0070 ARG kerekítő byte
0071-0072 Polinom kiért.(Ptr)
0073-0074 AUTO növekmény
0075 Grafikus terület kapcsoló
0076-0078 munkaterület
0079 DS$ érvényesség kapcsoló
007a-007b DS$ címe
007c-007d BASIC stack Pointer
007e-0080 munkaterület
0081 Parancs/Programm mód flag
0082 munkaterület
0083 HIRES/MULTI kapcsoló
0084 utolsó gr.utasítás színreg
0085 COLOR 2
0086 COLOR 1
0087-008a munkaterület
0088 PAINT mód kapcsoló
008c-008f munkaterület
0090 ST rendszerváltozó
0091 STOP kapcsoló
0092 munkaterület
0093 LOAD/VERIFY kapcsoló
0094 IEC Puffer kapcsoló
0095 IEC kimeneti Puffer
0096 munkaterület
0097 nyitott file-ok száma
0098 input-egység száma
0099 output-egység száma
009a KERNAL üzenetek kapcsolója
009b-009c SAVE Programkezdet
009d-009e Programvég
009f-00a2 munkaterület
00a3-00a5 Jiffy-clock
00a6-00a7 munkaterület
```

```
9b74 Szöveg beolvasása
9be0 Szövegmutató a szöveghez
9bda Füzérek összeláncolása
9c1b Füzér átmásolása
9c48 BASIC FRESTR
9caa Szöveghutató törlés a veremből
9cbb BASIC CHR$
9ccf BASIC LEFT$
9d03 BASIC RIGHT$
9d15 BASIC MID$
9d46 Szöveg és paraméter beolvasása
9d61 BASIC LEN
9d67 Szöveghparaméterek beolvasása
9d70 BASIC ASC
9d81 Egy byte-os érték X-be olvasása
9d93 BASIC VAL
9dd2 Cím és egy byte-os adat olvasása
9dde Cím beolvasása a $14/$15-be
9dfa BASIC PEEK
9e12 BASIC POKE
9e1b BASIC DEC
9e6a BASIC WAIT
9e87 FAC=ARG-FAC
9e95 FAC és ARG kitevők hasonlítása
9e9b FAC=(A/Y)+FAC
9eae FAC=ARG+FAC
9f7b FAC mantissza invertálás
9f93 LOG konstansok
9fb4 BASIC LOG
9fcc FAC=(A/Y)*FAC ROM-ból
9fe2 FAC=0.5*FAC
9fe6 FAC=(A/Y)+FAC ROM-ból
9fe9 FAC=(A/Y)-FAC ROM-ból
a00c FAC=(A/Y)/FAC ROM-ból
a012 FAC=(A/Y)*FAC
a07b FAC=ARG*FAC
a0dc ARG=(A/Y) ROM-ból
a10c ARG=(A/Y) RAM-ból
a1c2 FAC=FAC*10
a1   FAC=FAC*10
a1d4 FAC=(A/Y)/FAC
a1e7 FAC=ARG/FAC
a21f FAC=(A/Y)
a24c Accu#4=FAC
a24f Accu#3=FAC
a255 Változo=FAC
a291 FAC=ARG
a291 ARG=FAC
a2b0 FAC kerekítés
a2b0 FAC előjele
a2be SGN
a2d4 BASIC ABS
a2ed (A/Y) és FAC összehasonlítás
a2fd A FAC egésszé alakítása
a350 BASIC INT
a3a4 ASCII lebegőpontossá alakítása
a444 Lebegőpontos/ASCII konstansok
a45a A hiba sorszám kiírása
a456 IN szöveg
a45b A sorszám kiírása
a45f A/X két byte-os egész kiírása
a4bf FAC/ASCII konverzió
a5a9 32-bites 2-es komplemens számok
a5ac A TI/TI$ átalakítás konstansai
a5e4 BASIC SQR
a5ea FAC=ARG↑(A/Y)
a5ee FAC=ARG↑FAC
a627 FAC=-FAC
a632 EXP konstansai
a6a0 BASIC EXP
a6b3 A1*X+A2*X↑3+A3*X↑5.. kiértékelés
a6c9 A0+A1*X+A2*X↑2... kiértékelés
a6fd RND konstansok
a707 BASIC RND
a760 BASIC Stack $7C-ről $3D-re
a769 BASIC Stack $3D-ről $7C-re
a772 BASIC Stack növelése Y-nal
a77d Hibakezelés I/O rutinokhoz
a785 BASIC OPEN hívás
a78b BASIC CHROUT
a791 BASIC CHRIN
a797 BASIC CKOUT
a7a6 BASIC CHKIN
a7af BASIC GETIN hívás
a7b5 BASIC SYS
a7de BASIC SAVE
a7f0 BASIC VERIFY
a7f3 BASIC LOAD
a84d BASIC OPEN

00a8 IEC input byte
00a9 munkaterület
00aa IEC bitszámláló
00ab file-név hossza
00ac logikai file#
00ad másodlagos cím
00ae egység#szám
00af-00b0 filenév-Ptr
00b1 munkaterület
00b2-00b3 LOAD Programkezdet
00b4-00b5 Programvég
00b6-00b7 szalagPuffer-mutató
00b8-00c1 munkaterület
00c2 RVS ON flag
00c3 sorvég INPUT-nál
00c4-00c5 INPUT sor/oszlop
00c6 billentyűkód
00c7 INPUT CR-flag
00c8-00c9 kurzor címe
00ca kurzor oszlopa
00cb '"' flag
00cc munkaterület
00cd kurzor sora
00ce munkaterület
00cf INS kapcsoló
00d0 billentyűzet/képernyő flag
00d1-00e8 szabad terület
00e9 Grafika munkaterület
00ea-00eb 00c8-00c9 szín-párja
00ec-00ed billentyű dekód-mátrix
00ee akt. billentyű mátrix-kód
00ef jelek száma bill.Pufferben
00f0-00f6 munkaterület
00f7 szalag 1./2. kiírás flag
00f8 szalag-címke típusa
00f9 párhuzamos/soros IEC flag
00fa munkaterület
00fb aktuális ROM-térkép
00fc-00fd RS-232 munkaterület
00fe kurzorkezelés munkaterület
00ff-010f FAC/ASCII konverzió mt
0110-0112 regiszter másolatok
0113-0122 színbillentyűk kódjai
0123-01ff Processzor-stack
0200-0258 INPUT Puffer
0259-025a előző BASIC sor sor#
025b-025c CONT-cím
025d-027b munkaterület
027c-02ac diszk-Parancs Puffer
02ad-02b0 Pixel-kurzor koord.ák
02b1-02f5 BASIC munkaterület
02f6-02ff nem használt?
********  BASIC vektorok  ******
0300 $8686 melegindítás
0302 $8712 egy sor beolvasása
0304 $8956 tokenizálás
0306 $8b6e listázás
0308 $8bd6 utasítás értelmezés
030a $9417 kifejezés kiértékelés
030c $896a user utas.tokenizálás
030e $8b88 user-token listázás
0310 $8c8b user-token végrehajt.

********  KERNAL vektorok  *****
0312 $ce42 raszter interrupt
0314 $ce0e IRQ
0316 $f44c BRK
0318 $ef53 OPEN
031a $ee5d CLOSE
031c $ed18 CHKIN
031e $ed60 CKOUT
0320 $ef0c CLRCH
0322 $ebe8 CHRIN
0324 $ec4b CHROUT
0326 $f265 STOP
0328 $ebd9 GETIN
032a $ef09 CLALL
032c $f44c melegindítás
032e $f04a LOAD
0330 $f1a4 SAVE

0332-03f2 szalagPuffer
03f3-03f4 szalag outp. számláló
03f5-03f6 szalag input számláló
03f7-0436 RS232 Puffer
0437-0472 szalag inP.hibatábla
```

```
a85a BASIC CLOSE
a86b LOAD és SAVE Paraméterek
a897 ',' és Paraméter beolvasás X-be
a8a5 Következő byte ellenőrzése
a8b0 OPEN Paraméterek beolvasása
a8f8 A DS$ érvénytelenítése
a906 Füzér helyfoglalás
a954 Garbage Collection
aa70 BASIC COS
aa77 BASIC SIN
aac0 BASIC TAN
aaec Trigonometriai konstansok
ab1a BASIC ATN
ab8f BASIC RENUMBER
adca BASIC FOR
ae5a BASIC DELETE
aef7 BASIC USING
b386 BASIC INSTR
b42b BASIC TRAP
b440 BASIC RESUME
b4be BASIC ERR$
b507 BASIC HEX$
b544 BASIC PUDEF
b557 BASIC DO
b5ac BASIC EXIT
b603 BASIC LOOP
b652 BASIC TRON
b655 BASIC TROFF
b65b BASIC MID$ utasítás
b6cd BASIC AUTO
b6e8 BASIC HELP
b70c Hibás sor kiírása
b729 BASIC KEY
b7a7 KEY és sorszám, értékadás
b849 BASIC SOUND
b8bd BASIC VOL
b8d1 BASIC PAINT
b911 Színezés
b9d4 BASIC CHAR
bae2 BASIC BOX
bc56 Szögfüggvények számítása
bd35 BASIC GSHAPE
be29 BASIC SSHAPE
bf79 BASIC RGR
bf85 BASIC RCLR
bf87 BASIC RLUM
bfc1 BASIC JOY
bffd BASIC RDOT
c01e BASIC CIRCLE
c0d5 Egyenes húzása
c1a5 Pont megjelenítése
c1f3 Pont beolvasása
c21a Mező színezése
c264 Pont címe, offsetek
c289 Grafika bitmaszk
c291 Pont címének kiszámítása
c2ad Karakteres Pozíció számítása
c2d3 SCALE transzformáció
c2f0 Koordináták összege/különbsége
c2f4 összeg/különbség A/Y szerint
c2f6 Koordináták összege
c305 Koordináták kivonása
c322 Koordináta-különbség
c37b A PC (Pixel Cursor) feltöltése
c3b6 Szín-Paraméter beolvasása
c3d9 KoordinátaPár beolvasása
c49f Koordináta beolvasása
c4af Multicolor szín bitminták
c4b0 SIN táblázat
c4d9 BASIC DRAW
c50f BASIC LOCATE
c51a BASIC COLOR
c567 BASIC SCNCLR
c5b8 BASIC SCALE
c5c3 BASIC GRAPHIC
c637 Grafika kapcsoló ($83) táblázat
c63c Grafikus tárterület foglalása
c710 BASIC verem címmódosítások
c738 Grafikus terület törlése
c7ad BASIC verem 1 cím módosítása
c7bf Grafikus terület ellenőrzés
c7c9 Video alaphelyzetbe állítás
c7f0 Terület átrakása
c825 Szöveghivatkozások módosítása
c86b Szövegterület mérete
c877 Szöveg visszahivatkozások
c9bc BASIC DIRECTORY
c941 BASIC DSAVE
c951 BASIC DLOAD

0473 CHRGET belépési pontja
0473        inc $3b
            bne $0479
            inc $3c
0479 CHRGOT sei
            sta $ff3f ROM ki
            ldy #$00
            lda ($3b),y
            sta $ff3e ROM be
            cli
            cmp #$3a  >szám
            bcs $0493 igen, kész
            cmp #$20  blank?
            beq $0473 igen,ujra
            sec
            sbc #$30
            sec       carry on
            sbc #$d0  ha nem szám
0493        rts       kész

0494 RAM olvasó rutin A szerint
0494        sta $049c indirekt
            sei       cím beírás
            sta $ff3f ROM ki
049b        lda (**),y
            sta $ff3e ROM be
            cli
            rts       kész

04a5 RAM olv. $3b/3c vektorral
04b0 RAM olv. $22/23 vektorral
04bb RAM olv. $24/25 vektorral
04c6 RAM olv. $6f/70 vektorral
04d1 RAM olv. $5f/60 vektorral
04dc RAM olv. $64/65 vektorral

04e7-04ea PUDEF karakterek
04eb-04ee INSTR munkaterület
04ef hibakód
04f0-04f1 hibás sor sorszáma
04f2-04f3 TRAP-sor sorszáma
04f4 munkaterület
04f5-04f6 hiba címe
04f7 munkaterület
04f8-04f9 DO címe
04fa-04fb DO sorszáma
04fc-04ff hang időzítés értékek
0500      JMP     USR hívás
0501-0502 USR vektor ($991c)
0503-0507 utolsó RND érték
0508 memória inicializálás flag
0509-0512 logikai fileszám-tábla
0513-051c egységszám tábla
051d-0526 másodlagos cím-tábla
0527-0530 billentyűzet-Puffer
0531-0532 szabad RAM kezdőcím
0533-0534 szabad RAM teteje
0535 IEC timeout flag
0536 nem használt?
0537 munkaterület
0538 nem használt?
0539 szalagPuffer-mutató
053a nem használt?
053b karakter színe
053c FLASH ON flag
053d nem használt?
053e képernyő cím felső byte
053f billentyűPuffer mérete
0540 billentyű ismétlés kapcsoló
0541-0542 ismétlés számláló
0543 SHIFT/C=/CTRL kapcsoló
0544 munkaterület
0545-0546 billentyű dekód-vektor
0547 SHIFT/C= engedélyezés
0548 nem használt?
0549-0551 munkaterület
0552-0553 MONITOR PC
0554      MONITOR PSW (státusz)
0555      MONITOR A
0556      MONITOR X
0557      MONITOR Y
0558      MONITOR SP (stack-Ptr)
0559-055a nem használt?
055b-055c munkaterület
055d funkcióomb szöveghossz
055e akt. funkcióomb index
055f-0566 f.omb szöveghosszak
0567-05e7 f.omb szövegek
```

```
c968 BASIC HEADER
c99c BASIC SCRATCH
c[illegible] BASIC COLLECT
c9da BASIC COPY
c9f4 BASIC RENAME
ca00 BASIC BACKUP
ca16 Parancs kiadása
ca3f Parancs összeállítása
caf5 Diszk parancs szintaxis tábla
cb1f Diszk utasítás paraméterek
cc51 Egységszám beolvasása
cc69 File-név beolvasása
ccaa Paraméter kapcsoló ellenőrzése
cccf Diszk státusz beolvasás
cd40 Y/N beolvasása
cd57 DS$ érvénytelenítése
cd6e KEY 0, szöveg fordítva
cd74 A mint egész kiíratása
cd7f Programcím felső byte töltése
cd89 Copyright szöveg, $55-tel XOR
cdab Copyright szöveg kiiratása
cdb9 Üres ROM
****  BASIC INTERPRETER VÉGE  *******
ce00 Interrupt rutin
ceee Hang kikapcsolás maszk táblázat
cef0 Az óra (TI$) növelése
cf26 óra érték beolvasás
cf2d óra értékadás
cf36 MONITOR üzenetei
cf66 Üzenet kiírása
cf74 <CR> visszaírása (echo)
cf8a Színbilletyű kód olvasása
cf96 KERNAL tár olvasás $A1 szerint
cfa6 ROM-ok inicializálása
cfb3 KERNAL RAM olvasó rutin
cfbf PLAY gomb ellenőrzés, óra
cfea Paralel IEC(??) inicializálás
cff7 Üres ROM terület
d000 Karakter generátor
***  Upper-case (nagybetűk)  ********

D000..$D07F   @ABCDEFGHIJKLMNO
D080..$D0FF   PQRSTUVWXYZ[ <font> ]↑←
D100..$D17F   <szóköz> !"#$%&'()*+,-./
D180..$D1FF   0123456789 ,<=>?
D200..$D3FF   grafikus jelek

***  Lower-case (kisbetűk)  *********

D400..$D47F   @abcdefghijklmno
D480..$D4FF   PQRSTUVWXYZ[ <font> ]↑←
D500..$D57F   <szóköz> !"#$%&'()*+,-./
D580..$D5FF   0123456789:;<=>?
D600..$D67F   grafikus jel ABCDEFGHIJKLMNO
D680..$D6FF   PQRSTUVWXYZ grafikus jelek
D700..$D7FF   grafikus jelek

***  Magyar karakteres gépeken:  ****

D708..$D71F   ííô
D728..$D72F   ő
D758..$D75F   á
D760..$D767   é
D770..$D777   ü
D780..$D7C8   áÉÚÜóúóóÚó
D7D0..$D7D8   ű

d802 40-es szorzótábla képkezeléshez
d834 Képernyőméret megadása
d839 Kurzorpozíció írás/olvasás
d84e Video reset
d888 ESCAPE N
d88b Képernyő törlés
d89a HOME
d8a8 Kurzor címének kiszámítása
d8c1 Karakter beolvasása
d8ea Sor beolvasás képernyőről
d965 INPUT a képernyőről
d9ba Idézőjel figyelés
d9c7 Képernyő írásból visszatérés
d9d9 Kiírás a képernyőre
da21 Soremelés
da3d Sor mozgatása
da5e Sorbeszúrás
da89 Scroll (görgetés) fel
daf7 Képernyő sor törlés
db11 Billentyűzet olvasás
db70 Billentyűzet-mátrix olvasása
db7a Billentyűzet dekódolás
```

```
05e8-05eb munkaterület
05ec-05ef cartridge-ROM kapcs.
05f0-05f4 munkaterület
05f5-06eb nem használt?
06ec-07af BASIC-stack
07b0 szalag-kapcsoló (puff/tár)
07b1-07cc szalag munkaterület
07cd-07d8 RS232 munkaterület
07d9-07e4 RAM olvasás PHP/PLP-vèl
07e5 ablak alsó sora
07e6 ablak felső sora
07e7 ablak bal széle
07e8 ablak jobb széle
07e9 scroll engedélyezés kapcs.
07ea auto-insert flag (ESC+A)
07eb előző karakter
07ec munkaterület
07ed kurzor alatti szín
07ee-07f0 folytatósor bittábl.
07f1 munkaterület
07f2 BASIC SYS A pszeudoreg.
07f3           X pszeudoreg.
07f4           Y pszeudoreg.
07f5           ST pszeudoreg.
07f6-07f7 munkaterület
07f8 RAM/ROM olvasás kapcsoló
07f9 színbill. kód RAM/ROM-ból
07fa-07fc munkaterület
07fd PAL/NTSC óra korrekció
07fe-07ff nem használt?
*********  karakter-mód:  *****
0800-0bff SZÍN-mátrix
0c00-0fff KÉP-mátrix
1000-3ff5 BASIC terület (16K)
    -fcff                (64K)

*********  grafikus-mód:  *****
1000-17ff BASIC terület (16K)
1800-1bff SZÍN-mátrix
1c00-1fff FÉNYERŐ-matrix
2000-3fff BIT-mátrix

1000-17ff üres          (64K)
1800-1bff SZÍN-mátrix
1c00-1fff FÉNYERŐ-matrix
2000-3fff BIT-mátrix
4000-fcff BASIC-terület
```

```
dc41 Funkció gombok kódjai
dc49 Kiírás a képernyőre
dc8c <CR> a képernyőre
dc9b ESCAPE 'O'
dca7 Vezérlő karakterek ($00-$1F)
dcf1 Szín karakter
dcfa Kurzor jobbra
dd00 Kurzor le
dd0d Kurzor fel
dd1c Kurzor balra
dd27 CTRL/N
dd2e CTRL/H
dd35 CTRL/I
dd3e CTRL/SHIFT/N - CHR$(142)
dd47 Felső 128 karakter
dd56 Vezérlő karakterek ($80-$9F)
dd99 DELETE
ddce INSERT
de0e ESCAPE funkciók
de1a ESCAPE funkciók címei
de4c ESCAPE 'R'
de5e ESCAPE 'T'
de60 ESCAPE 'P'
de70 HOME & HOME
de8b ESCAPE 'I'
dea0 ESCAPE 'D'
deab ESCAPE 'O'
dee1 ESCAPE 'P'
def6 ESCAPE 'V'
df04 ESCAPE 'N'
df1d ESCAPE 'L'
df2d ESCAPE 'M'
df36 ESCAPE 'C'
df39 ESCAPE 'A'
df2f Kurzor alatti karakter olvasása
df39 Folytatósor ellenőrzés
df46 Folytatósor vizsgálás/törlés
df59 Folytatósor vizsgálás
df7a Bitmaszk a folytatósorokhoz
df82 ESCAPE 'T'
df95 ESCAPE 'K'
dfbb Kurzor jobbra
dfd4 Kurzor balra
dff6 Kurzorpozíció mentése
dfff SPACE kiírása
e001 Karakter kiírása
e211 Karakter visszaírása
e01e Billentyűzet táblázatok címei
e02c Normál billentyűk
e067 Shift billentyűk
e0a8 C= (Commodore) billentyűk
e0e9 Control billentyűk
e13a Szöveg a SHIFT/RUN billentyűhöz
e133 Szín vezérlő billentyűk
e143 Billentyűkhöz tartozó szinek
e153 TALK a soros IEC-n
e156 LISTEN a soros IEC-n
e177 Egy byte kivitele soros IEC-n
e1f7 SECONDARY LISTEN a soros IEC-n
e203 SECONDARY TALK a soros IEC-n
e21d OUTPUT a soros IEC-n
e22c UNTALK a soros IEC-n
e23d UNLISTEN a soros IEC-n
e252 INPUT a soros IEC-n
e2b8 IEC CLOCK vonal 1-re állítás
e2bf IEC CLOCK vonal 0-ra állítás
e2c6 IEC DATA vonal 1-re állítás
e2cd IEC DATA vonal 0-ra állítás
e2d4 IEC olvasás
e2dc 1 ms körüli késleltetés
e2ea 20 ms körüli késleltetés
e2f3 1 ms körüli érték a TIMER 2-be
e2fc 20 ms körüli érték a TIMER 2-be
e311 késleltetés
e319 PRESS...ON TAPE kiírása
e364 Szalag IRQ képkioltás
e378 Visszatérés kazetta kezelésből
e38d Kazetta motor bekapcsolás
e3a2 Copyright szöveg
e3b0 Kazetta motor kikapcsolása
e3b7 Puffer feltöltése SPACE-szel
e3c3 Puffer-mutató feltöltése
e3ce STOP figyelés kazettához
e3e4 IRQ lekezelése
e403 21 ms a TIMER 1-be
e413 Egy impulzus kiírása
e447 1 ms értéke
e452 0.25 ms értéke
e45d 0.5 ms értéke
```

```
******* TED regiszterek *******

$fd00 RS232 I/O regiszter
$fd01 RS232 státusz-regiszter
$fd02 RS232 kontroll-regiszter
$fd03 RS232 parancs-regiszter
.
.
$fd10 szalag-státusz
  bit 2  0 - PLAY ON
         1 -      OFF

$fd30 keyboard-port (oszlop-
      kijelölő regiszter)

$fdd0 BANK-port (ROM-enable)
  bit 0-1  00 - BASIC
           10 - C1 low
           11 - C2 low

  bit 2-3  00 - KERNAL
           10 - C1 high
           11 - C2 high

$fec0 paralel  IEC I/O
.
.
$fec5

$fef0 paralel  IEC I/O
.
.
$fef5

$ff00 timer1-low
$ff01 timer1-high
$ff02 timer2-low
$ff03 timer2-high
$ff04 timer3-low
$ff05 timer3-high
$ff06 video kontroll-regiszter
  bit 7  teszt
  bit 6  extended-color mód
  bit 5  HI-RES mód
  bit 4  képkioltás
  bit 3  24/25 sor mód
  bit 2-0  függőleges eltolás

$ff07
  bit 7  reverse mód
  bit 6  PAL/NTSC
  bit 5  TED ki/be
  bit 4  MULTI-mód
  bit 3  38/40 oszlop
  bit 2-0  vízszintes eltolás

$ff03 Billentyű-mátrix sor tár.

$ff09 Interrupt kontroll-reg.
  bit 7  IRQ (bármilyen)
  bit 6  timer3
  bit 5  -
  bit 4  timer2
  bit 3  timer1
  bit 2  fényceruza
  bit 1  raszter
  bit 0  -

$ff0a Interrupt engedélyező reg.
        (enable interrupt)
  bit 7  -
  bit 6  timer3
  bit 5  -
  bit 4  timer2
  bit 3  timer1
  bit 2  fényceruza
  bit 1  raszter
  bit 0  raszter reg. 8. bitje

$ff0b Raszter regiszter
        (sorszámláló)

$ff0c Kurzor pozíció
  bit 7-2: nem használt
  bit 1-0: pozíció 9-8 bitjei
```

```
e468 0 bit kiírása szala9ra
e474 1 bit kiírása szala9ra
e480 Byte kezdő jel kiírás
e48c E9y byte kiírása szala9ra
e4ba Adatblokk kiírása
e535 Puffer kiírása
e56c Fejléc kiírása
e5b0 Tár kiírása (SAVE szala9ra)
e5f0 EOT kiírása
e5fa Szala9 impulzusok se9édtáblázat
e5fd E9y impulzus beolvasása
e68e Bit beolvasás se9édtáblázat
e691 E9y bit beolvasása szala9ról
e6d5 Byte kezdet keresése
e6ec E9y byte beolvasása szala9ról
e74b E9y blokk beolvasása szala9ról
e8d3 Beolvasás a szala9Pufferba
e8f3 Beolvasás a tárba (LOAD szala9.)
e917 Se9édértékek szala9 olvasáshoz
e91d Blokk keresése
e9cc Szala9 file-név olvasás
ea21 TAPE kívánt fejléc keresése
ea5b RS-232 interrupt kezelés
eaf1 RS-232 e9y byte beolvasása
eb25 RS-232 CHKIN, CKOUT
eb37 RS-232 e9y byte kiírása
eb46 RS-232 OPEN, CLOSE
eb58 KERNAL üzenetek
ebc6 KERNAL üzenet kiírás
ebd9 KERNAL GETIN
ebe8 KERNAL CHRIN
ec24 Szala9 INPUT
ec4b KERNAL CHROUT
ec8b IEC INPUT
ec96 Paralel IEC INPUT
ecdf IEC OUTPUT
ece6 Paralel IEC OUTPUT
ed18 KERNAL CHKIN
ed3a IEC CHKIN
ed57 Szala9 CHKIN
ed60 KERNAL CKOUT
ed85 IEC CKOUT
eda1 Szala9 CKOUT
eda9 Paralel IEC ellenőrzés
edd4 Paralel IEC se9édrutinok
edfa KERNAL IEC TALK
ee02 Paralel IEC TALK
ee1a KERNAL IEC SECONDARY TALK
ee21 Paralel IEC SECONDARY TALK
ee2c KERNAL IEC LISTEN
ee34 Paralel IEC LISTEN
ee4d KERNAL IEC SECONDARY LISTEN
ee54 Paralel IEC SECONDARY LISTEN
ee5d KERNAL CLOSE
eeca Törlés a file táblázatból
eee8 File létének ellenőrzése
eef8 File Paraméterek táblázatból
ef08 KERNAL CLALL
ef0c KERNAL CLRCH
ef23 KERNAL IEC UNLISTEN
ef2a Paralel IEC UNLISTEN
ef3b KERNAL IEC UNTALK
ef42 Paralel IEC UNTALK
ef53 KERNAL OPEN
f043 KERNAL LOAD
f0f0 Szala9 LOAD
f160 SEARCHING FOR... kiírása
f189 LOADING/VERIFYING kiírása
f194 KERNAL SAVE
f1b5 IEC SAVE
f211 IEC CLOSE
f228 SAVING kiírása
f234 Szala9 SAVE
f265 KERNAL STOP
f273 I/O ERROR kiírások
f2a4 RESET rutin
f2ce KERNAL vektorok inicializálása
f2eb KERNAL vektor táblázat
f2eb  42ce      $ce42, Raszter IRQ
f2ed  0ece      $ce0e, IRQ
f2ef  4cf4      $f44c, BRK utasítás
f2f1  53ef      $ef53, KERNAL OPEN
f2f3  5dee      $ee5d, KERNAL CLOSE
f2f5  18ed      $ed18, KERNAL CHKIN
f2f7  60ed      $ed60, KERNAL CKOUT
f2f9  0cef      $ef0c, KERNAL CLRCH
f2fb  e8eb      $ebe8, KERNAL CHRIN
f2fd  4bec      $ec4b, KERNAL CHROUT
f2ff  65f2      $f265, KERNAL STOP
```

```
$ff0d Kurzor Pozíció 7-0 bitjei

$ff0e oszcillátor1 frekvenciája

$ff0f oszcillátor2 frekvenciája

$ff10
  bit 1-0: oszc#2 frekv.9-8 bit

$ff11 Han9 vezérlő re9iszter
  bit 7: Sound reload
  bit 6: oszc#2 zaj-jel bekaPcs.
  bit 5: oszc#2 né9yszö9jel be
  bit 4: oszc#1 bekaPcsolás
  bit 3-0: han9erő

$ff12 Video-kontroll re9iszter
  bit 7-6: -
  bit 5-3: Bit-térkéP helye
  bit 2: Karakter-9en. ROM/RAM
  bit 1-0: oszc#1 frekv. 9-8 bit

$ff13 Video-kontroll re9iszter
  bit 7-2: Karakter 9en. helye
  bit 1: ha 1, nincs f0 sor-
         visszafutás közben
  bit 0: ROM en9edélyezés jelző

$ff14 Video-mátrix kijelölő re9.
  bit 7-3: Video-mátrix helye
  bit 2-0: -

$ff15 Háttérszín#0

$ff16 Háttérszín#1

$ff17 Háttérszín#2

$ff18 Háttérszín#3

$ff19 Keretszín

$ff1a Aktuális karakterPozíció
  bit 7-2: -
  bit 1-0: sorkezdet 9-8 bitjei

$ff1b Aktuális sor kezdő kar.Poz

$ff1c Videosor felső bit
  bit7-1: -
  bit 0: sorszámláló 8. bit

$ff1d Aktuális video-sor

$ff1e Vízszintes Pozíció 8-1. bit

$ff1f Video vezérlés
  bit 7: -
  bit 6-3: villo9ás-időzítő
  bit 2-0: karakteren belüli
           sorszámláló

$ff3e ROM enable

$ff3f ROM disable
```

```
f301  d9eb      $ebd9, KERNAL GETIN
f303  08ef      $ef08, KERNAL CLALL
f305  4cf4      $f44c, KERNAL WARM
f307  4af0      $f04a, KERNAL LOAD
f309  a4f1      $f1a4, KERNAL SAVE
f30b I/O chipek inicializálása
f338 TED inicializálás értékei
f352 Memória teszt és törlés
f3d2 Funkció9omb szöve9hosszak
f3da Funkció9ombokhoz rendelt szöve9
f40c KERNAL SETNAM
f413 KERNAL SETLFS
f41a KERNAL SETMSG
f41c KERNAL STATUS
f423 KERNAL IEC Time-out
f427 KERNAL RAM teteje
f436 KERNAL RAM kezdete
f445 MONITOR
f478 MONITOR 'R'
f4d7 MONITOR 'M'
f50a MONITOR ';'
f529 MONITOR '>'
f54b MONITOR 'G'
f570 MONITOR utasítások
f580 MONITOR utasítás-vektorok
f59a E9y sor (8 byte) dumP-olása
f5ce MONITOR 'C'
f5d1 MONITOR 'T'
f60e MONITOR 'H'
f66e MONITOR file-kezelés (L,S,V)
f70a MONITOR 'F'
f724 MONITOR 'D'
f752 E9y sor disassemblálása
f83d Assembly/Disassembly táblázat
f91f MONITOR 'A'
fa7d Hexa számje9y ellenőrzés
fa8b Kétje9yű hexa szám olvasása
faa0 Számértékké alakítás
faab Paraméter beolvasás
fafb Cím kiírása
fb08 SPACE kiírása
fb0b '?' kiírása
fb10 E9y byte-os hexa szám kiírása
fb20 Hexa/ASCII konverzió
fb35 Kurzor fel kiírása
fb3c <CR> kiírása
fb3f E9y byte olvasása a Pufferből
fb5b Aktuális cím betöltése
fb64 Feldol9ozandó byte-ok száma
fb72 Számláló csökkentése
fb86 Vé9cím csökkentése
fb94 Aktuális cím növelése
fba0 Kezdő és vé9cím beolvasása
fbb7 Processzor re9iszterek mentése
fbc1 Re9iszterek visszatöltése
fbcb STOP 9omb ellenőrzés
fbd8 Szöve9 kiírása
fc00 Közös ROM-terület
fc19 I/O modul kezdőcíme
fc1e ROM-ok létének ellenőrzése
fc56 CBM szöve9
fc59 ROM inicializálások me9hívása
fc7b ROM térkeP se9édtáblázat
fc7f E9y byte olvasása másik ROM-ból
fc89 Szubrutin hívása másik ROM-ban
fcb3 Interrupt rutin
fcc9 ROM térkép váltása
fcd1 Üres ROM terület
fcf1 átkaPcsoló szubrutinok
fd00 I/O terület
ff00 TED re9iszterek
ff40 Üres ROM terület
ff49 Kie9észítő KERNAL rutinok
ff81 KERNAL u9rótáblázat

FF81  JMP  $D84E    Video-RESET
FF84  JMP  $F30B    I/O inicializálás
FF87  JMP  $F352    RAM teszt
FF8A  JMP  $F2CE    I/O inicializálás
FF8D  JMP  $F2D3    I/O vektor beállítás
FF90  JMP  $F41A    Státusz beállítás
FF93  JMP  $EE4D    Másodla9os cím LISTEN
FF96  JMP  $EE1A    Másodla9os cím TALK
FF99  JMP  $F427    RAM teteje
FF9C  JMP  $F436    RAM kezdete
FF9F  JMP  $DB11    Billentyűzet olvasás
FFA2  JMP  $F423    IEC busz time-out
FFA5  JMP  $EC8B    IEC inPut
FFA8  JMP  $ECDF    IEC outPut
```

```
FFAB  JMP  $EF3B    UNTALK
FFAE  JMP  $EF23    UNLISTEN
FFB1  JMP  $EE2C    LISTEN
FFB4  JMP  $EDFA    TALK
FFB7  JMP  $F41C    Státusz behozatal
FFBA  JMP  $F413    SETLFS
FFBD  JMP  $F40C    SETNAM
FFC0  JMP ($0318)   $EF53 OPEN
FFC3  JMP ($031A)   $EE5D CLOSE
FFC6  JMP ($031C)   $ED18 CHKIN
FFC9  JMP ($031E)   $ED60 CKOUT
FFCC  JMP ($0320)   $EF0C CLRCH
FFCF  JMP ($0322)   $EBE8 BASIN
FFD2  JMP ($0324)   $EC4B BSOUT
FFD5  JMP  $F043    LOAD
FFD8  JMP  $F194    SAVE
FFDB  JMP  $CF2D    óra állítása
FFDE  JMP  $CF26    óra olvasása
FFE1  JMP ($0326)   $F265 STOP billentyű
FFE4  JMP ($0328)   $EBD9 GET
FFE7  JMP ($032A)   $EF08 CLALL
FFEA  JMP  $CEF0    Az idő növelése
FFED  JMP  $D834    Sor/oszlopszám
FFF0  JMP  $D839    Kurzor Pozíció
FFF3  JMP  $FC19    I/O kezdőcím
fff6 RESET rutin
************ ROM vége ****************
```

L. FELHASZNÁLT ÉS AJÁNLOTT IRODALOM A COMMODORE 16-HOZ

[1] dr.úry László: COMMODORE C-16
LSI Alkalmazástechnikai Tanácsadó Szolgálat, Budapest 1985

[2] dr.úry László: COMMODORE 64
LSI Alkalmazástechnikai Tanácsadó Szolgálat, Budapest 1984

[3] COMMODORE 16 BEDIENUNGSHANDBUCH
COMMODORE Büromaschinen GmbH

[4] COMMODORE 16 Felhasználói Kézikönyv
NOVOTRADE RT. 1985

[5] Bevezetés a BASIC-nyelvbe
NOVOTRADE RT. 1986

[6] Tóth Viktor: A Commodore 16-os belső felépítése
NOVOTRADE RT. 1986

[7] Erdős Iván: COMMODORE PLUS/4, C-16, C-116 ROM-lista
LSI Alkalmazástechnikai Tanácsadó Szolgálat, Budapest 1986